

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

«На правах рукопису»
УДК 004.932

«До захисту допущено»
Завідувач кафедри
_____ О.Л. Тимошук
«__» _____ 20__ р.

**Магістерська дисертація
на здобуття ступеня магістра
зі спеціальності 124 Системний аналіз
на тему: «Інтелектуальна система мерчандайзингу. Детекція та
розпізнавання асортименту»**

Виконав:
студент II курсу, групи КА-62м
Левчук Святослав Богданович _____

Керівник:
к.т.н., доц.
Дідковська М.В. _____

Рецензент:
к.т.н., доц.
Заболотня Т.М. _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____

Київ
2018

РЕФЕРАТ

Магістерська дисертація: 126 с., 47 рис., 30 табл., 2 додатки, 31 джерело.

Об'єктом дослідження є методи мерчандайзингу товарів в торгових точках.

Предметом дослідження є методи класифікації товарів на полицях в торгових точках.

Мета роботи – розробка інтелектуальної системи мерчендайзингу, яка дозволить зменшити використання людського ресурсу та максимально оптимізувати процес мерчендайзингу за рахунок автоматичного моніторингу наявності товару на полицях та розробка системи класифікації товарів як складової частини системи мерчандайзингу для аналізу товарів на полиці відносно планogram магазину.

В роботі розглянуто і проаналізовано сучасні системи мерчандайзингу та їх недоліки, також, розглядаються існуючі методи класифікації.

Запропоновано метод класифікації товарів в магазині з спеціально розробленою згортковою нейронною мережею, який побудовано на основі методів з використанням згорткових нейронних мереж, з нелінійними класифікаторами та адаптивним методом оптимізації.

Інтелектуальна система мерчандайзингу та система класифікації асортименту реалізовані за допомогою мови програмування Python з використанням СУБД MySQL.

Результати даної роботи рекомендується використовувати для моніторингу якості викладки товарів на полицях та контролю наповненості полиць у торгових точках.

МЕРЧАНДАЙЗИНГ, КЛАСИФІКАЦІЯ, РИТЕЙЛ, ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ, БІНАРНИЙ КЛАСИФІКАТОР, ТОВАРНА ПОЛКА

ABSTRACT

Master thesis explanatory note: 126 p., 47 fig., 30 tab., 2 appendices, 31 sources.

The object of research – intelligent merchandising system.

The subject of research – classification methods of goods on shelves in stores.

The purpose of the work is to develop an intelligent merchandising system that will reduce the use of human resources and maximize the process of merchandising through automatic monitoring of the availability of goods on shelves and to develop of goods classification system as a part of a merchandising system for the analysis of goods on the shelf in relation to the store planograms.

In the work, modern merchandising systems and their shortcomings are considered and analyzed, as well as existing classification methods are considered.

Goods classification method with specially developed convolutional neural network, which is constructed on the basis of methods using convolutional neural networks, with nonlinear classifiers and an adaptive optimization method is proposed.

Intelligent merchandising system and assortment classification system are implemented using Python programming language with MySQL DB.

The results of this work are recommended for monitoring the compliance with the planogram and availability of the goods on shelves in stores.

MERCHANDAIZING, CLASSIFICATION, RETAIL, CONVOLUTIONAL NEURAL NETWORKS, BINARY CLASSIFICATION, STORE SHELF

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП.....	10
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	13
1.1 Аналіз актуальності задачі інтелектуальної системи мерчендайзингу	13
1.2 Аналіз існуючих систем мерчендайзингу	16
1.3 Аналіз існуючих підходів до класифікації зображень	20
1.3.1 Аналіз методу контрольованої класифікації.....	22
1.3.2 Аналіз методу неконтрольованої класифікації	23
1.3.3 Аналіз алгоритмів класифікації зображень	24
1.4 Постановка задачі розробки інтелектуальної системи мерчендайзингу та системи класифікації товарів на полицях.....	29
Висновки за розділом 1	30
РОЗДІЛ 2. МАТЕМАТИЧНІ ОСНОВИ ЗАДАЧІ КЛАСИФІКАЦІЇ.....	31
2.1 Аналіз базової структури CNN.....	31
2.1.1 Згорткові шари в CNN.....	32
2.1.1.1 Приклад роботи згорткового шару.....	34
2.1.1.2 Використання багатьох фільтрів	38
2.1.1.3 Заповнення нулями	38
2.1.3 Об'єднуючі шари	39
2.1.4 Повнозв'язні шари	40
2.1.5 Аналіз структури вхідних зображень	41
2.1.6 Поняття тензора	41
2.1.7 Процес навчання	43
2.2 Аналіз запропонованої структури нейронної мережі для класифікації товарів в ритейлі	45
2.3 Опис шарів запропонованої CNN.....	46
2.4 Аналіз використаних функцій активації.....	47
2.4.1 Нелінійна функція ReLU (Rectified linear units).....	47

	7
2.4.2 Функція активації сигмоїд	49
2.5 Аналіз функції Dropout.....	50
2.6 Аналіз функції ціни (похибки) для задачі бінарної класифікації	51
2.7 Аналіз запропонованого модифікованого адаптивного методу стохастичного градієнтного спуску	55
2.7.1 Алгоритм.....	56
2.7.2 Правило оновлення кроку	57
2.7.3 Корекція μ і σ	58
2.8 Процес підготовки даних для навчання запропонованої мережі	59
2.8.1 Підготовка початкового набору зображень	60
2.8.2 Розбиття фотографій на навчальну, валідаційну і тестову вибірки....	61
2.8.3 Урізноманітнення зображень за допомогою функцій деформації.....	61
2.8.4 Приведення зображень до необхідного розміру	62
2.9 Аналіз процесу навчання і тестування запропонованої мережі	62
2.10 Використання запропонованого класифікатора	63
Висновки за розділом 2	64
РОЗДІЛ 3. АНАЛІЗ АРХІТЕКТУРИ ПРОГРАМНОГО ПРОДУКТУ	65
3.1 Обґрунтування вибору платформи та мови програмування.....	65
3.2 Обґрунтування вибору архітектури класифікатора	67
3.2.1 Архітектура глибокого мультикласового класифікатора на базі класичних згорткових мереж.....	67
3.2.2 Модифікована мережа з модулем Inception	71
3.2.3 Запропонована архітектура з бінарним класифікатором	76
3.3 Аналіз архітектури бази даних	77
3.4 Аналіз архітектури програмного продукту	80
Висновки за розділом 3	81
РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ.....	83
4.1 Аналіз роботи алгоритму класифікації.....	83
4.2 Аналіз якості роботи системи класифікації	84

	8
4.2.1 Аналіз якості навчання	85
4.2.2 Аналіз якості класифікації	87
4.3 Аналіз якості роботи інтелектуальної системи мерчандайзингу	90
4.4 Керівництво користувача	96
Висновки за розділом 4	101
РОЗДІЛ 5. СТАРТАП	103
5.1 Опис ідеї проекту	103
5.2 Технологічний аудит ідеї проекту	105
5.3 Аналіз ринкових можливостей запуску стартап–проекту	106
5.4 Розроблення ринкової стратегії проекту	112
5.5 Розроблення маркетингової програми стартап–проекту	115
Висновки за розділом 5	118
ВИСНОВКИ	120
ПЕРЕЛІК ПОСИЛАНЬ	123
ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДОПОВІДІ	127
ДОДАТОК Б ЛІСТИНГ	141

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ТТ – Торгова точка

РЦ – Розподільчий центр

OSA – On Shelf Availability

OOS – Out of Shelf

ІТ – Інформаційні технології

НМ – Нейронні мережі

БД – База даних

СУБД – Система управління базами даних

CNN – Згорткова нейронна мережа

DCNN – Глибока згорткова нейронна мережа

ВСТУП

Магазин сьогодні – це не тільки полиці з товаром і каса на виході. Це скоріше завершальний елемент у складному ланцюжку проходження продукції від виробника до кінцевого споживача. Покупцеві в ній видно тільки вітрину магазину. За фасадом залишається робота логістів, комірників, мерчендайзерів, маркетологів, бухгалтерів і менеджерів різних напрямів. Величезна кількість товару проводиться, фасується, вантажиться, доставляється, викладається на полиці магазинів і нарешті продається. За всім цим кругообігом товару в природі потрібен нагляд. І людині не під силу впоратися без різних допоміжних елементів – датчиків, штрих-кодів, магнітних міток – всього того, що допомагає автоматизувати облік переміщення товару.

Неодноразові дослідження фахівців доводять, що в середньому 50% роздрібних покупок товарів споживчого призначення не плануються покупцями заздалегідь. Найбільша ефективність впливу на рішення про покупку доводиться в момент покупки в торговому залі магазину. Вже протягом перших 5–7 секунд після того, як покупець бачить товар, він приймає рішення про його покупку [1].

Можна сказати, що хороший мерчендайзинг замінює продавця–консультанта. Правильна викладка, як невидимий продавець, буквально підштовхує клієнта до здійснення покупки. Створювався і вперше застосовувався мерчендайзинг саме в магазинах самообслуговування. Коли він зарекомендував себе як ефективний механізм продажів, то на нього почали звертати увагу маркетологи і фахівці в області продажів.

В результаті мерчендайзинг став в ряд з маркетингом і продажами. Вважається, що продажі вирішують короткострокові питання збуту продукції, тобто тут і зараз. Маркетинг вирішує питання довгострокові, тобто містить заходи, які допомагають залучати клієнтів через якийсь час. Мерчендайзинг же займається вирішенням середньострокових завдань реалізації продукції [2].

Оскільки на сьогоднішній день активно проводиться автоматизація та оптимізація робочих та бізнес процесів в галузі роздрібної торгівлі, то мерчендайзинг є слабкою ланкою, адже для моніторингу якості та повноти викладки товару на полиці все ще залучається людський ресурс.

Важливою складовою системи мерчендайзингу є система класифікації товарів на полиці, так як для аналізу наявності та правильності викладки товару на полиці необхідно визначити, що за товар знаходиться на тій чи іншій позиції на полиці і чи місце на полиці не є порожнім. Для цього необхідно класифікувати об'єкт, який знаходиться на певному місці на полиці. Після класифікації стане можливим визначення правильності розташування і наявності товарів на полиці, що сприяє продажам.

Саме тому ціль роботи заключається в розробці інтелектуальної системи мерчендайзингу, яка дозволить зменшити використання людського ресурсу та максимально оптимізувати процес мерчендайзингу за рахунок автоматичного моніторингу наявності товару на полицях та розробці системи класифікації товарів на полиці як складової частини системи мерчендайзингу для аналізу товару на полиці відносно планогам магазину.

Для досягнення мети розв'язані такі задачі:

- Аналіз ринку ритейла та існуючих рішень.
- Аналіз алгоритмів та методів класифікації зображень.
- Розробка алгоритму класифікації товарів.
- Розробка методу класифікації товарів на полицях.

Об'єктом дослідження є методи мерчендайзингу товарів в торгових точках.

Предметом дослідження є методи класифікації товарів на полицях в торгових точках.

Наукова новизна полягає в запропонованому підході до автоматизації мерчендайзингу, що поєднує в собі систему класифікації і систему сегментації та мапінгу; запропонованому підході до класифікації товарів, що базується на спеціально розробленій глибокій згортковій нейронній мережі для визначення відповідності розташованого товару на полиці до планограми.

Практичним результатом є реалізація інтелектуальної системи мерчендайзингу у вигляді програмного продукту, що дозволяє проводити аналіз полиць в реальному часі з точністю 96% та дозволяє накоплювати історію наповненості полиць для подальшого статистичного аналізу, а також реалізація системи класифікації товарів як окремого модулю інтелектуальної системи мерчендайзингу, яка дозволяє класифікувати товари на полиці згідно планограми і є невідємною частиною загальної системи.

Робота складається з 5 розділів. В першому розділі розглядаються сучасні системи мерчандайзингу та їх недоліки, також, розглядаються існуючі методи класифікації. В другому розділі описується запропонований метод класифікації товарів, проаналізовані методи, з яких він складається та наведено алгоритм роботи на прикладній задачі. В третьому розділі проводиться аналіз системи з точки зору архітектури та програмної реалізації. В четвертому розділі аналізується якість роботи запропонованої інтелектуальної системи мерчандайзингу. В п'ятому розділі проводиться аналіз програмного продукту як стартапу.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз актуальності задачі інтелектуальної системи мерчандайзингу

Криза 2015–2016 років найбільше торкнулась галузі споживання. За 2 роки оборот роздрібної торгівлі скоротився на 15%, фактично ми відкотилися до рівня 2010–2011 років. Падіння реальних доходів населення позначилося на розвитку роздрібної торгівлі: воно триває третій рік і неминуче змінює споживчу поведінку. Для прикладу споживачі все більше уваги звертають на проведення промоактивностей.

Таке падіння ринку дало поштовх для розвитку та впровадження ІТ в галузь роздрібної торгівлі (ритейл). Продукти та системи ІТ почали активно використовуватися для автоматизації та оптимізації існуючих бізнес процесів, зменшення використання людської праці в рутинній монотонній роботі. До таких продуктів та систем можна віднести:

- Системи лояльності – аналізують кожний чек та пропонують промоакції та спецпропозиції для конкретного сегменту покупців або ж навіть для окремого покупця.
- Системи ціноутворення – базуючись на ціновій еластичності та ціновій політиці мережі, надають рекомендації для ціноутворення, з ціллю максимально підвищити прибутковість або ж залучити максимально широку аудиторію.
- Системи прогнозування продажів – аналізують дані за певний історичний період та моделюють майбутні продажі з врахування промоактивностей. Вони дозволяють зменшити складські залишки та збільшити доступність товару на полиці.
- Системи автозамовлення – базуючись на прогнозі, страховому запасі, презентаційному запасі та інших факторах, розраховують необхідне замовлення від постачальника на РЦ та розподіляють товар з РЦ на ТТ.

Як видно з основних систем, що використовуються в ритейлі, ключовими задачами для розвитку мереж є:

- забезпечення доступності товару на полиці;
- розширення аудиторії покупців;
- зменшення списання товарів та товарних залишків;
- встановлення оптимальних цін.

Якщо більшість із задач можна вирішити аналізом історичних даних та моделюванням майбутніх подій, то доступність товару на полиці забезпечити та проконтролювати не так просто. Чому це так? Доступність товару на полиці залежить від багатьох факторів:

- доставка товару від постачальника;
- доставка товару з РЦ;
- коректний прогноз продажів;
- виставлення товару в ТТ на полиці;
- контроль наявності товару в ТТ на полиці;
- контроль умов розміщення товару в ТТ на полиці.

Перші три задачі можна автоматизувати, проте три останні на даний момент часу зазвичай виконують люди, а саме мерчендайзери. Саме від цих людей залежить чи є товар на полиці і чи у належній він якості. Це впливає не тільки на прибуток ТТ а і на довіру до мережі та збереження аудиторії покупців, оскільки від розміщення товару на полиці залежить якість проведення промоактивностей та якість історичних даних для аналізу.

Для оцінки наявності товару на полиці існує показник OSA – On Shelf Availability, за підвищення якого так чи інакше борються всі учасники ринку. Даний показник вимірюється у відсотках і є оберненим до такого показника, як OOS – Out of Shelf (показник втрачених продажів). Тут варто зазначити, що навіть працівники галузі роздрібної індустрії часто плутають події Out of Stock і Out of Shelf, тим більше що у них однакова аббревіатура (OOS). Дуже важливий момент, що підкреслює відмінність даних показників, полягає в тому, що, згідно зі статистикою, близько 50% випадків Out of Shelf (а саме Out of Shelf є показником

неможливості придбання товару покупцем) відбувається при ненульових залишках (тобто коли подія out of shelf вже настало, а події out of stock – ще немає) [3].

Трохи статистики від ECR (ECR – «Efficient Consumer Response») – міжнародна некомерційна організація «Ефективний відгук на запити споживачів»), що характеризує важливість показників OOS і OSA для постачальників і ритейлерів:

- в середньому всього 20% позицій в статусі OOS поповнюється протягом 8 годин;
- 8,3%, середній показник OOS по індустрії в світі, не змінюється останні 8 років;
- підвищення OSA на 3% принесе ~ 1% інкрементального (зі зростаючим трендом) зростання обсягів поставок для постачальника;
- підвищення OSA на 2% принесе ~ 1% інкрементального (зі зростаючим трендом) зростання обсягів продажів для ритейлера;
- у 47% випадків при відсутності потрібного йому товару на полиці покупець не зробить покупку в цьому магазині;
- у 46% випадків при відсутності на полиці магазину потрібного йому товару покупець не зробить покупку товару даного виробника [1].

Виходячи з вище сказаного очевидно, що своєчасне виявлення нехватки товару на полиці та якнайшвидше його поповнення є однією із критичних задач для сучасного ритейлера. І саме тому використання автоматизованої інтелектуальної системи мерчендайзингу є надзвичайно актуальним для сьогоденного ритейлера.

1.2 Аналіз існуючих систем мерчендайзингу

Задачі мерчендайзингу в сфері роздрібної торгівлі виконуються протягом багатьох років. Були вироблено та впроваджено безліч підходів, які проходили випробовування часом або були неефективними. На даний момент на українському ринку роздрібної торгівлі можна виділити три основні системи мерчендайзингу:

1. Перша система є найпростішою та найдешевшою. Вона базується повністю на людській праці. Її суть заключається в наступному: постачальник або магазин виділяє людину, відповідальну за розміщення певного товару або заповнення полки. Дана система має право на життя в невеликих магазинах, але в великих супермаркетах з постійним потоком клієнтів вона далеко не ідеальна, оскільки необхідно занадто багато людей для контролю, або ж якщо це мерчендайзери постачальників можливі конфлікти конкурентів.
2. Друга система є більш поширеною для торгових мереж, але також базується на людському ресурсі. Вона заключається в наступному: компанія посередник, або постачальник товарів, або ритейлер розробляють програмний продукт для мобільного телефону або планшету, далі працівник зі встановленим застосунком фотографує полиці з товаром. Застосунок оброблює фотографії, проводить сегментацію та мапінг товарів і аналізує доступність товару на полицях. Після аналізу ТТ автоматично формується звіт, що відправляється відповідальному за ТТ для подальших дій. Даний метод є значно кращий за попередній але все одно не дозволяє повністю автоматизувати процес мерчендайзингу.

На ринку такими системами являються:

- ShelfMatch – розробником рішення являється російська компанія Kuznech. Це програма для автоматизованого порівняння

планограмм з фактичним розташуванням продукції на полицях магазинів. Технологія дозволяє контролювати розміщення десятків тисяч товарних найменувань. З фотографії або відеозапису полиці з товарами ShelfMatch автоматично визначає, які товари присутні в викладенні, чи відповідає викладка планограмі, і створює аналітичний звіт. Аналіз викладки проводиться автоматично виключно на основі вхідних (НЕ промаркованих текстом) зображень і не вимагає додаткових калькуляцій. Звіт передається в підрозділи компанії (або точково співробітникам) в електронному вигляді за допомогою мобільного додатку. Відповідь від системи приходить в реальному часі, а значить, співробітник може внести зміни в викладку відразу ж, на місці;

- Intelligence retail – ще одна російська розробка. Принцип роботи цього рішення аналогічний попередньому. В основі лежать технології SAP. Розробники за'являють, що по фотографії система може за 10 секунд розпізнати 96% SKU, знайти відсутні товари, визначити кількість наявних, знайти ціnnики і порівняти ціни на них з рекомендованими постачальником, проаналізувати відповідність викладки планограмі і т.д. Система дозволяє скоротити час аудиту одного магазину на 80%, і як наслідок мерчендайзер може обходити на третину більше точок в день. Точність даних зростає на 25% в порівнянні з ручною обробкою даних;
- Trax Retail, ST–Mobi, Polka та інші. Також є досить сильними гравцями на ринку. Всі вони мають в основі одну й ту ж саму ідею.

На превеликий жаль дані системи залежать від людського ресурсу і не можуть працювати в режимі реального часу. Для того щоб отримати звіт або аналітику необхідно, щоб працівник сфотографував всі полки та відправив фото на обробку. Цей процес вимагає багато часу особливо для мереж супермаркетів та великих мереж. Також одна з проблем це підробка фотографії або заміна її не

актуальну, оскільки система довіряє саме працівнику, який робить та завантажує фото, вона не має можливості перевірити наскільки фото достовірне та відповідає реальності.

3. Третій тип системи зараз вважається революційним, це так звані “розумні полиці”. Дана система заключається в наступному: в полиці вмонтовуються спеціальні датчики, які вимірюють вагу, температуру полиці. На базі отриманих даних системою робиться висновок про наповненість полиці базуючись на планограмі. Дана система реагує на зміну температури (наприклад в холодильниках) та повідомляє про це відповідних працівників, щоб уникнути псування товару. Недоліком даної системи є неможливість аналізу відповідності товару на полиці відносно асортименту планограми та значно вища вартість в порівнянні з попередніми системами. Для прикладу можна навести Smart Shell Solution. Дана система дає можливість співробітникам компаній спостерігати з мобільних пристроїв за тим, що відбувається на полицях, помічати порушення планограм та стежити за продажами. Також дана система може автоматично формувати замовлення. Система задовільняє 80% потреб в "розумному" ритейлі задовольняються за рахунок базових функцій: контроль відсутності продукції, спостереження за температурою та наявністю електропостачання в холодильниках. Використовуються датчики для зняття показників крутного моменту і віддаленого зняття показників за рахунок зчитування магнітних полів. Таким чином, відкривши систему і вибравши будь-який магазин на карті, працівник компанії отримує показники працеспроможності пристрою, його розташування, продаж і так далі. Система може бути незалежним джерелом аналітики статистики та інтегруватися в будь-яку існуючу ПО, таку як SAP. Полиці можуть «самі» замовити продукцію та розсилати повідомлення за попередньо встановленими подіями.

На превеликий жаль система нездатна визначити коректність розміщення товару на полиці відносно планограми. Дана система призначена більше для постачальника, оскільки може надати повну інформацію про умови зберігання товару, що дозволяє постачальнику контролювати як ритейлер виконує свої зобов'язання перед ним. Для ритейлера дана система менш продуктивна, так як необхідно будувати складні планограми з врахуванням ваги товару для полиці та розбивати великі секції полиць на більш дрібні для точнішого виявлення нестачі товару. Порівняльна характеристика систем мерчендайзингу наведена в Таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика систем мерчендайзингу.

Тип системи	Перший тип	Другий тип	Третій тип
Автоматизованість (по 5 бальній шкалі)	1	3	5
Вартість підтримки (по 5 бальній шкалі)	1	3	5
Вартість впровадження (по 5 бальній шкалі)	1	2	5
Складність впровадження (по 5 бальній шкалі)	3	3	3

Продовження таблиці 1.1

Тип системи	Перший тип	Другий тип	Третій тип
Складність підтримки (по 5 бальній шкалі)	5	2	1

Як видно за Таблиці 1.1 на сьогоднішній день важко досягнути балансу між вартістю, автоматизованістю та складністю підтримки системи мерчендайзингу. Хоча можна сказати що другий тип є найбільш оптимальним для досягнення балансу.

1.3 Аналіз існуючих підходів до класифікації зображень

Одним з основних завдань обробки та аналізу зображень є класифікація, тобто віднесення зображення до того чи іншого класу в залежності від його особливостей, наприклад, наявність 4 лап і хвоста у котів.

У інтелектуальній системі мерчендайзингу класифікація є одним із двох основних етапів. На цьому етапі раніше сегментовані зображення товарів подаються на вхід класифікатору, який в свою визначає до якого класу відносяться об'єкти. Так як класифікація являється одним з основних етапів в процесі визначення відповідності асортименту на полицях, до системи є певні вимоги:

- висока точність класифікації;
- здатність працювати при певних викривленнях зображення;
- швидкість опрацювання зображень;
- повний автоматизм роботи.

Виходячи зі сформульованих вимог до системи дамо визначення автоматичної системи для класифікації і проведемо аналіз найпоширеніших методів класифікації з метою виявлення їх слабких та сильних сторін.

Автоматичною системою класифікації об'єктів можна вважати програмно-апаратну структуру для візуального розпізнавання об'єктів і подальшого розподілу по попередньо визначених класах. Зазвичай система автоматичної класифікації об'єктів складається з наступних елементів:

1. бази даних, яка містить заздалегідь визначені шаблони, які порівнюються з об'єктом для класифікації до відповідної категорії;
2. обчислювального пристрою;
3. зовнішніх сенсорів.

Задача класифікації товарів на полицях в торгових точках є частковим випадком задачі класифікації зображень. Тому варто розглянути основні методи та алгоритми для вирішення задачі класифікації зображень.

Основними методами класифікації зображень є:

- контрольована класифікація;
- неконтрольована класифікація.

Основними алгоритмами класифікації зображень є:

- класична нейронна мережа (ANN);
- наївний Байєсівський класифікатор;
- K найближчих сусідів;
- машини опорних векторів (SVM);
- дерево рішень;
- згорткова нейронна мережа.

Детально дані підходи описані нижче.

1.3.1 Аналіз методу контрольованої класифікації

В контрольованій класифікації деякі пікселі є завідомо згруповані і задають мітки класів. Даний процес відомий як навчання. Опісля класифікатор використовує навчені пікселі для класифікації інших зображень. Це потребує додаткової інформації, яка збирається аналітиком перед процесом тестування. На цьому етапі аналітик визначає навчальні набори, які представляють кожний клас, а також алгоритм генерує межі для прийняття рішень. Найчастіше на практиці використовуються наступні методики контрольованої класифікації - паралелепіпед, мінімальна відстань до середнього та максимальна правдоподібність [5].

В методах контрольованої класифікації зазвичай виділяють наступні кроки:

1. Визначення аналітиком навчальних вибірок для кожного інформаційного класу.
2. Виділення критеріїв (середнє значення, варіація, коваріація, тощо).
3. Класифікація всіх пікселів.
4. Побудова карти інформаційних класів.

Переваги контрольованої класифікації:

1. Помилки можуть бути виявлені операторами і виправлені [6].
2. Потрібна експертиза, як наслідок – даний підхід дає точний результат.

Недоліки контрольованої класифікації:

1. Не підходить для роботи з Big Data, так як кожна область даних потребує свого експерта.
2. Дуже часоємнісний – потребує багато часу для попередньої ідентифікації класів.

1.3.2 Аналіз методу неконтрольованої класифікації

У неконтрольованій класифікації пікселі групуються з використанням їх унікальних властивостей. Цей процес відомий як кластеризація, а утворені групи називаються кластерами. Аналітик в даному підході вирішує скільки кластерів має бути. Неконтрольована класифікація використовується у випадках, коли немає підготовлених наборів тренувальних пікселів. У неконтрольованій класифікації попередня інформація не потрібна. Така класифікація не потребує втручання людини, вона повністю автоматизована. Алгоритм автоматично визначає кластери в даних, а аналітик тільки дає їм відповідні назви.

В методах контрольованої класифікації зазвичай виділяють наступні кроки:

1. Кластеризація даних.
2. Класифікація всіх пікселів на базі кластерів.
3. Побудова спектральної карти класів.
4. Найменування кластерів аналітиком.
5. Побудова карти інформаційних класів.

Переваги неконтрольованої класифікації:

1. Аналітик витрачає менше часу на процес класифікації. В результаті тільки потрібні зображення класифікуються.
2. Підходить для Big Data.

Недоліки неконтрольованої класифікації:

1. Ніякі види тренування не використовуються, тому метод потребує серйозних знань в предметній області, або знань про метод, який варто застосувати для області.
2. Для великих об'ємів даних час виконання різко зростає. Так створюється неефективний класифікатор.

1.3.3 Аналіз алгоритмів класифікації зображень

Основними кроками в алгоритмах вирішення задачі класифікації є [7]:

1. Отримання зображень для обробки.
2. Препроцесинг зображення – процес перетворення зображення до вигляду, який машина зможе ефективно розпізнати (трансформація, усунення шумів, корекція, тощо).
3. Виявлення та витяг важливих ознак.
4. Навчання.
5. Класифікація.

Основними алгоритмами являються:

1. Класичні нейронні мережі (ANN).

Класична нейронна мережа (ANN) - це тип штучного інтелекту, який має обмежується деякими базовими функціями людського мозку. Нейронні мережі накопичують експериментальні знання. Будь-яка мережа складається з набору послідовних шарів, а кожен шар складається з набору нейронів. Всі нейрони кожного шару пов'язані зваженими з'єднаннями з усіма нейронами на попередніх і наступних шарах [8].

Нейронну мережу можна розглядати як зважений направлений граф, в якому вузли є нейронами, а ребра з вагами – зв'язками. Кожний нейрон обчислює зважену суму всіх вхідних сигналів і генерує результуючий вихідний сигнал, який базується на функціях активації. Функції бувають кусочно-лінійні, сигмоїдні, гауссові та інші. Мережа складається з одного вхідного шару, одного вихідного шару, і в залежності від необхідності може включати в себе приховані шари. Кількість елементів на вихідному шарі рівна кількості інформаційних класів, тоді як кількість елементів на вході дорівнює розмірності кожного пікселя. Більшість джерел визначає класичну нейронну мережу, як повнозв'язну нейронну мережу прямого поширення зі зворотнім поширенням помилки.

У фазі навчання мережа навчає ваги ітераційно за допомогою тренувальних наборів з даними. Мережа дає вихід, який відповідає кожному входу, потім згенеровані вихідні дані порівнюються з бажаним вхідними. Помилка між входом і виходом використовується для навчання ваг. Процедура навчання закінчується, коли помилка стає меншою за попередньо визначену порогову величину. Після цього починається процес класифікації даних [9].

Використання класичних нейронних мереж є неефективним підходом через зазвичай велику розмірність вхідного вектора, велику кількість нейронів прихованих шарів і як наслідок великими затратами на обчислювальних ресурсів на навчання і обчислення.

Наведемо приклад. Розмір вхідного зображення 25 на 25 пікселів, це означає 625 нейронів на вхідному шарі. Кількість нейронів прихованих шарів вибирається в 10 – 20 разів більше чим у вхідному. Нехай нейронна мережа складається з 2 приховані шари, тоді перший прихований шар буде мати 6000 нейронів (3,75 мільйона зв'язків з вхідним шаром), другий шар 3000 нейронів (18 мільйонів зв'язків з першим прихованим шаром). Кожний зв'язок між нейронами має вагу, для якої обчислюється градієнт похибки. Ці дані дозволяють наочно зрозуміти наскільки задача розпізнавання є складною для класичних нейронних мереж. Ще одним недоліком є неможливість враховувати особливості розташування одних пікселів відносно інших на зображенні.

2. Наївний Байєсівський класифікатор.

Наївний Байєсівський класифікатор на представленні імовірності. Він присвоює клас, який має найбільшу оцінку наступної ймовірності, вектору властивостей отриманому з ROI. Цей процес є оптимальним, якщо атрибути ортогональні. Проте даний класифікатор працює добре і без цього твердження. Простота методу дозволяє забезпечити ефективну роботу при невеликих навчальних вибірках. Будуючи імовірнісні моделі класифікатор є стійким до викидів, а також він дозволяє створювати не строгі границі для прийняття рішення, це допомагає уникнути перенавчання. Крім того, це створює м'які межі прийняття рішень, результатом чого є уникнення перенавчання. Проте,

довільний вибір варіанту моделі розподілу для оцінки імовірностей $P(x)$ разом з відсутністю гнучкості для вихідних рішень приводить до обмеженої продуктивності для складних мультикласових конфігурацій класифікатора [10].

3. K найближчих сусідів.

Класифікатор k-найближчих сусідів (KNN) відсікає гіперсфери в просторі екземплярів шляхом передачі основного класу k-найближчих екземплярів відповідно до визначеної метрики. Цей класифікатор є асимптотично-оптимальним, а також його реалізація дозволяє проводити швидкі тести [11]. Проте в цьому методі присутні багато недоліків. Класифікатор дуже чутливий до змін розмірностей [12]. Збільшення розмірності приводить до зменшення простору ознак, а локальні однорідні області, які являються прототипами різних класів, поширюються. Ефективність класифікації напряму залежить від обраної метрики. Більше того, невелике значення k призводить до появи хаотичних границь і робить процес класифікації дуже нестабільним відносно викидів.

4. Мащини опорних векторів.

Ядра машин опорних векторів (SVM) неявно відображають вектори вхідних ознак у простір з більшою розмірністю за допомогою функції ядра з Гаусівською шириною. У перетвореному просторі будується максимально гіперболічна гіперплощина з врахуванням двокласової проблеми. Дві паралельні гіперплощини будуються симетрично по обидві сторони гіперплощини, яка розділяє дані. Ціль полягає в використанні відстані між двома зовнішніми розділеними гіперплощинами, дана відстань називається границею. Зроблено заяву про те, що покращена маржа, покращиться помилка спрощення класифікатора існуватиме. SVM відрізняються від інших класифікаторів підходом до мінімізації структури ризику, який спрямований на мінімізацію верхньої межі узагальненої помилки, тоді як більшість класифікаторів спрямовані на мінімізацію емпіричного ризику - помилки на тренувальній вибірці. Алгоритм SVM є направлений на пошук функції рішення, яка мінімізує функціонал. SVM надають можливість навчати нелінійні класифікатори у високорозмірних просторах, використовуючи хвилинні тренувальні набори. Це

дозволено під час вибору поділу векторів (які називаються опорними векторами), що характеризує правильні границі між класами. Таким чином даний підхід є одним з найбільш ефективних підходів до класифікації [13].

5. Древа рішень.

Алгоритм дерева рішень базується на ієрархічному методі і використовує непараметричний підхід [14]. Дерево рішень обчислює приналежність до класу шляхом багаторазового поділу набору даних на рівномірні підмножини. Ієрархічний класифікатор дозволяє приймати та відхиляти мітки класів на кожній проміжній стадії. Цей алгоритм включає в себе 3 етапи:

1. Поділ вузлів.
2. Пошук термінальних вузлів.
3. Розподіл класів по термінальних вузлах.

Даний алгоритм є частково подібним на алгоритми нейронних мереж.

6. Згорткові нейронні мережі.

Згорткова нейронна мережа (CNN) була представлена французьким дослідником Яном Лекуном в 1998 році [15]. Дані мережі являються мережами з прямою передачею сигналу, в них сигнал передається тільки від входу до виходу. Ідея згорткової нейронної мережі полягає в чередуванні згорткових (convolutional) і об'єднуючих (pooling) шарів, а також наявності одного або декількох повнозв'язних шарів на виході.

Стандартна структура мережі зображена на рисунку 1.1:

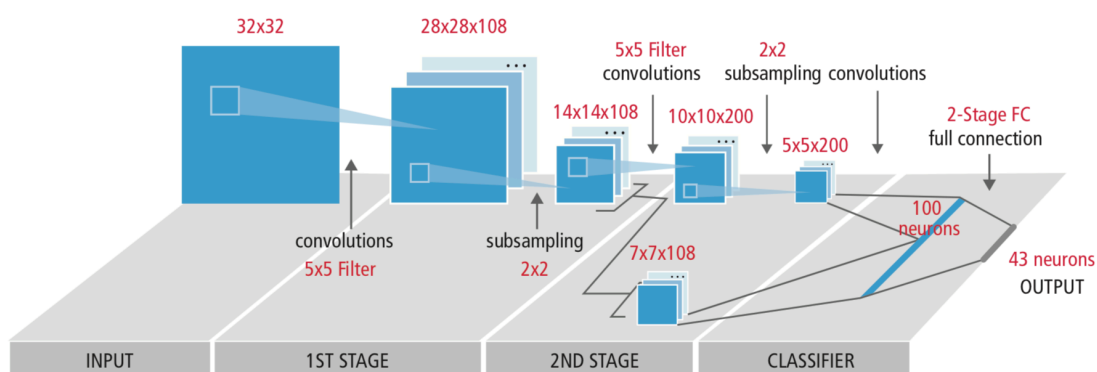


Рисунок 1.1 – Структура згорткової нейронної мережі

Така архітектура включає в себе три основні парадигми:

1. Локальне сприйняття.
2. Роздільні ваги.
3. Об'єднання.

Під локальним сприйняттям мається на увазі, що на вхід одного нейрона подається не все зображення, а тільки певна його область. Такий підхід дозволяє зберігати топологію зображення від шару до шару (що неможливо при використанні класичних мереж).

Концепція роздільних ваг вказує, що для великої кількості зв'язків використовується невелика кількість ваг. Так при вхідному зображенні розміром в 28 на 28 пікселів, кожний з нейронів наступного шару прийме зображення розміром всього 4 на 4 пікселі, причому кожний з елементів буде опрацьований одним і тим же набором. Легко помітити, що для 10 наборів розміром 4 на 4 кількість зв'язків буде приблизно 200 тисяч, в порівняння з 18 мільйонами для класичних мереж, а кількість налаштовуваних параметрів буде всього 200.

Процес обмеження ваг покращує узагальнюючі властивості мережі (генералізація) і дозволяє реагувати саме на зміни у зображенні, а не на сторонні шуми [16].

Попри те, що така структура запропонована в більшості літературних джерел, сьогодні запропоновано багато покращень і допрацювань, які дозволяють стати згортковим нейронним мережам найкращим підходом до класифікації зображень.

Аналіз підходів до класифікації показує, що всі вони не позбавлені недоліків. Тому вибір того чи іншого методу залежить від конкретного завдання.

Для вирішення завдання класифікації товарів на полицях, яке являє собою частковий випадок завдання класифікації зображень найкраще використовувати згорткові нейронні мережі. В останні роки такий підхід до класифікації дуже швидко розвивається, на найпрестижнішому міжнародному конкурсі по класифікації зображень ILSVRC (ImageNet Large Scale Visual Recognition

Challenge) з 2012 року перемагають тільки згорткові нейронні мережі. Також такий підхід дозволяє зробити класифікатор інваріантним до викривлень зображень, а також дозволяє легко донавчати класифікатор. Тому доцільно використати саме цей підхід до вирішення задачі класифікації.

1.4 Постановка задачі розробки інтелектуальної системи мерчендайзингу та системи класифікації товарів на полицях

Метою дипломної роботи є розробка та дослідження інтелектуальної системи мерчендайзингу і системи класифікації товарів як її невід'ємної частини, що дозволить підвищити доступність товару на полицях та виявлення пропусків та невідповідності на полках відносно планogram. Проблема наявності та викладки товару на полицях є надзвичайно актуальною для ритейлу сьогодення. Відсутність товару та неправильна викладка на полиці зумовлює не лише втрату продажів, а й відтік покупців до конкурента. Сучасні системи мерчендайзингу намагаються автоматизувати процес мерчендайзингу, але все ще не можуть відмовитися від людського ресурсу, що час від часу призводить до помилок. Повна автоматизація дозволить підвищити точність аналізу реального розміщення товару на полиці відносно планogramи.

Тому в рамках цієї задачі необхідно:

По-перше, дослідити та провести аналіз існуючих систем мерчендайзингу.

По-друге, дослідити та провести аналіз існуючих систем класифікації товарів.

По-третє, розробити систему класифікації товарів на полицях.

По-четверте, розробити систему мерчендайзингу.

По-п'яте, реалізувати вищесказані системи у вигляді єдиного програмного продукту.

Висновки за розділом 1

У даному розділі було розглянуті та описані типові системи мерчендайзингу. Проаналізовано їх сильні та слабкі сторони.

Розглянуто та детально досліджено системи мерчендайзингу конкурентів. Виявлено та аргументовано їх недоліки.

Досліджено системи, що існують на ринку, їх сегмент клієнтів та можливість застосування на ринку України.

Проаналізовано підходи до класифікації товарів на полицях в торгових точках, як часткового випадку задачі класифікації зображень. Виділено їх недоліки та переваги. Описано основні принципи їх роботи.

Показано актуальність та перспективність розробки системи мерчендайзингу, на основі чого сформульовано постановку задачі дипломної роботи та виділено етапи її розв'язку.

РОЗДІЛ 2. МАТЕМАТИЧНІ ОСНОВИ ЗАДАЧІ КЛАСИФІКАЦІЇ

Опишемо основні математичні підходи, що використовуються в запропонованому методі класифікації товарів, розглянемо запропонований метод класифікації та алгоритм його використання як складову інтелектуальної системи мерчандайзингу.

2.1 Аналіз базової структури CNN

Існує багато підходів до класифікації зображень, проте найефективнішою сьогодні є класифікація з використанням згорткових нейронних мереж.

CNN являються нейронними мережами прямого поширення. В них дані рухаються строго від входу до виходу. Архітектура CNN буває декількох варіантів. Проте в загальному випадку мережа складається з шарів згортки та шарів об'єднання, які об'єднані в групи, і одного чи декількох повнозв'язних шарів перед виходом. Групи згорткових і об'єднуючих шарів накладаються один на одний, створюючи форму глибокої моделі. На рисунку 2.1 зображено стандартну структуру глибокої згорткової нейронної мережі, що використовується для задачі класифікації зображень (чи фотографій).

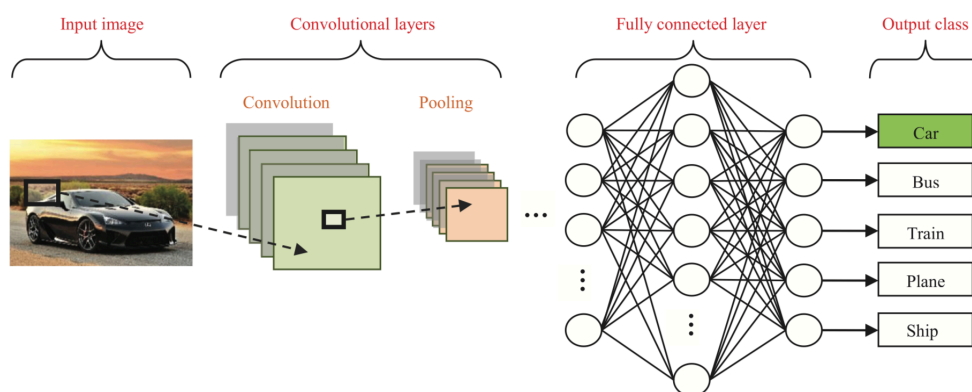


Рисунок 2.1 – Стандартна структура DCNN для класифікації

Стандартний процес класифікації в CNN виглядає наступним чином:

1. На вхід мережі подається зображення, яке необхідно класифікувати.
2. Зображення проходить обробку декількома групами згорткових та об'єднувальних шарів.
3. Дані, отримані на виході пункту 2 подаються на один чи більше повнозв'язних шарів.
4. Останній повнозв'язний шар виводить назву класу, до якого належить зображення.

Попри те, що описана архітектура являється базовою, в реальних задачах розпізнавання її необхідно допрацьовувати. Це дозволяє зменшити необхідні обчислювальні ресурси, а також підвищити точність роботи мережі.

Дальше опишемо основні елементи CNN, які впливають на процес класифікації.

2.1.1 Згорткові шари в CNN

Першим і основним шаром в CNN являється згортковий. Мережі можуть мати довільну кількість таких шарів.

Згорткові шари використовуються для витягування унікальних ознак зображення. Вони вивчають як представлені унікальні ознаки для кожного вхідного зображення. Нейрони в згорткових шарах згруповані в карти ознак. Кожний нейрон має рецептивне поле, яке зв'язане з сусідніми нейронами попереднього шару за допомогою ваг [17]. Входи з обчисленими вагами згортуються для обчислення нової карти ознак, а результати згортки передаються за допомогою нелінійної функції активації. Ваги всіх нейронів в одній карті ознак є однакові, проте різні карти ознак на одному згортковому шарі мають різні ваги, тому декілька ознак можуть бути отримані на кожній локації [15,17]. Формально, k -ий вихід Y_k карти ознак може бути обрахований за наступною формулою:

$$Y_k = f(W_k \cdot x) \quad (2.1)$$

де вхід позначається як x , згортковий вектор, зв'язаний з k -ою картою ознак як W_k , а множення в даному випадку відноситься до двовимірного згорткового оператора, а f являє собою нелінійну функцію активації [18].

Графічно процес обробки вхідного зображення розміром $32 \times 32 \times 3$ в згортковому шарі зображено на рисунку 2.2.

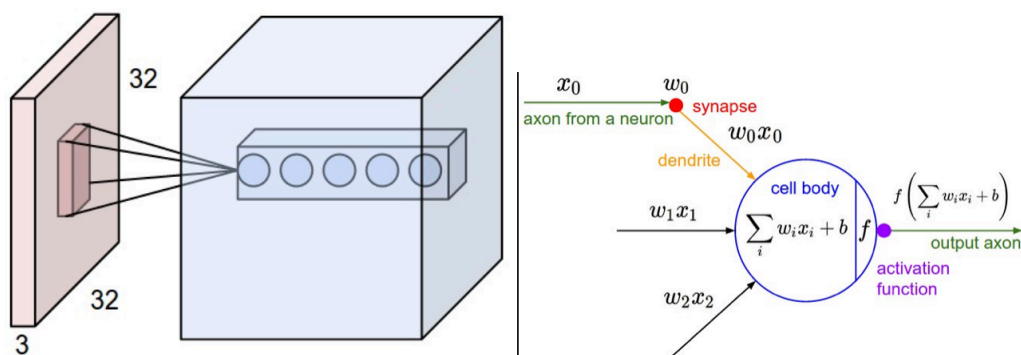


Рисунок 2.2 – Приклад обробки даних в згортковому шарі

Вхідне зображення розміром $32 \times 32 \times 3$ зображено червоним. Кожний нейрон згорткового шару з'єднаний з локальним регіоном на зображенні. На рисунку 2.2 п'ять нейронів згорткового шару сприймають один регіон зображення. Вихід кожного нейрона вираховується так, як показано в правій частині на рисунку 2.2.

2.1.1.1 Приклад роботи згорткового шару

Процес, який відбувається у згортковому шарі представлено графічно на рисунках 2.3 – 2.8:

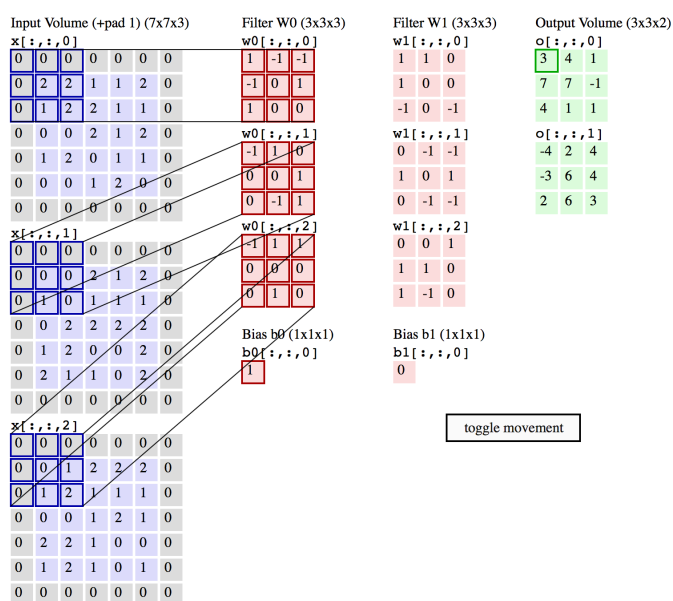


Рисунок 2.3 – Графічний приклад роботи згорткового шару

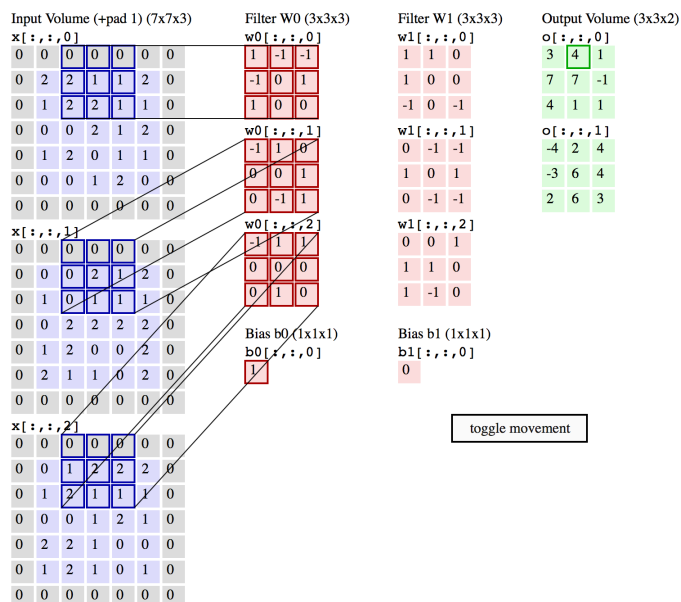


Рисунок 2.4 – Графічний приклад роботи згорткового шару

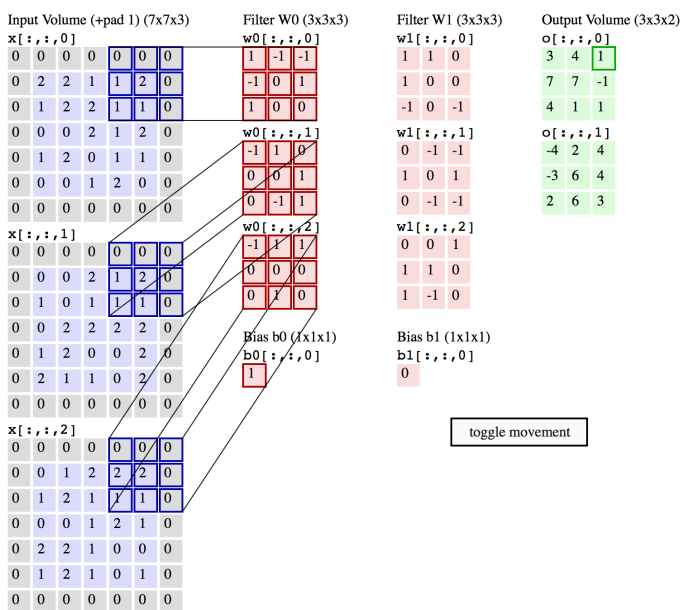


Рисунок 2.5 – Графічний приклад роботи згорткового шару

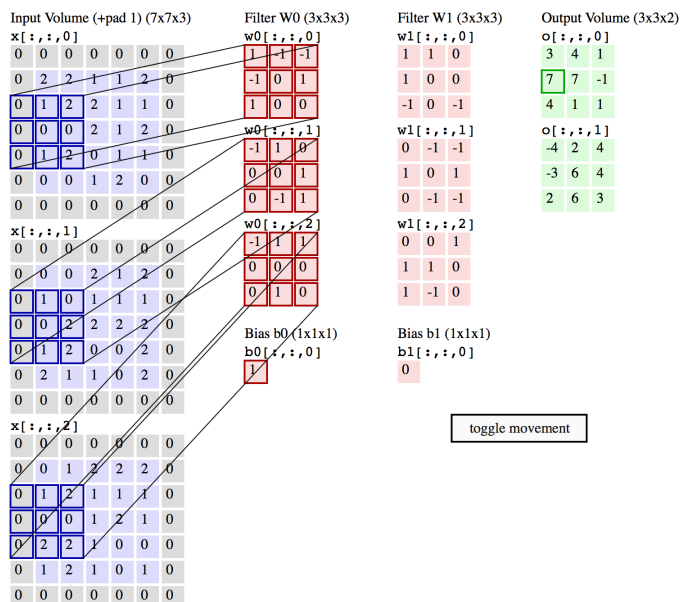


Рисунок 2.6 – Графічний приклад роботи згорткового шару

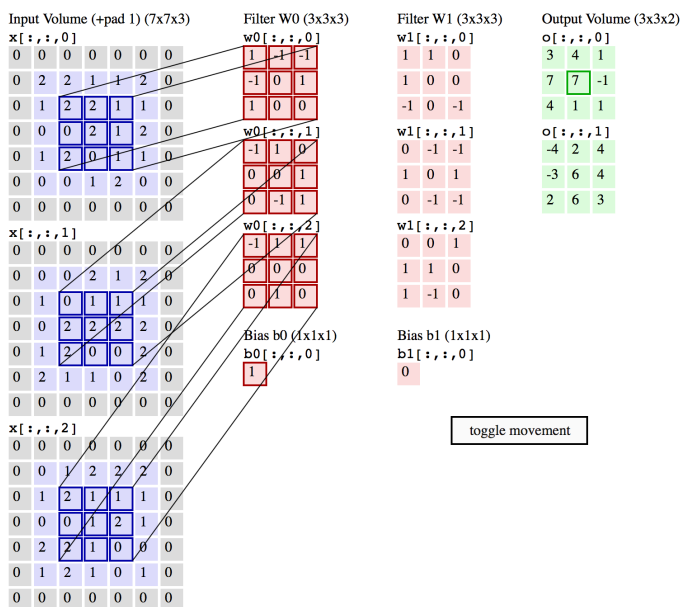


Рисунок 2.7 – Графічний приклад роботи згорткового шару

Дальше процес руху фільтрів є аналогічним. Кінцевий стан зображено на рисунку 2.8.

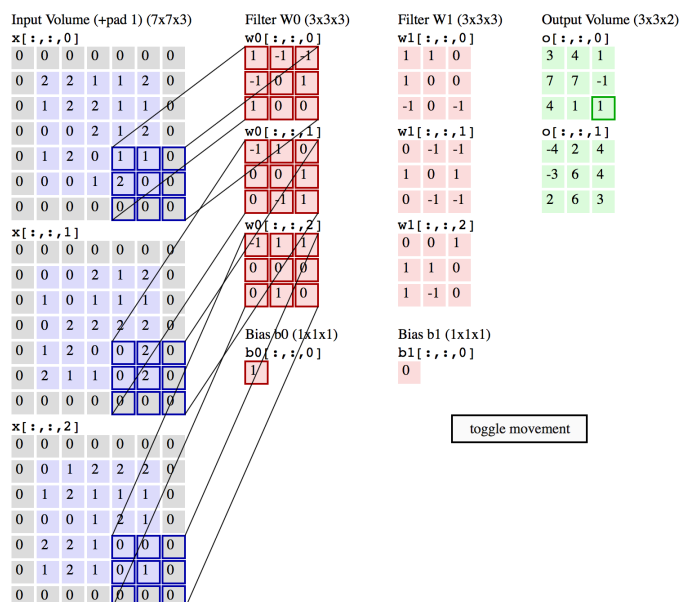


Рисунок 2.8 – Графічний приклад роботи згорткового шару

На рисунках 2.3 – 2.8 синім кольором зображено вхідне зображення. Кожна матриця відповідає з один канал (за червоний, зелений, синій кольори).

Фільтри ваг зображені червоним, а вихідна матриця зеленим. На вхід подається матриця розміром 5x5x3. Згортковий шар має наступні параметри:

- два фільтри;
- розмірність фільтру 3x3;
- крок рівний двом;
- відступ рівний одиниці.

Вихідний шар має розмірність $(5 - 3 + 2)/2 + 1 = 3$. Відступ застосовується до вхідного зображення заповнюючи його краї нулями. На рисунках 2.3 – 2.8 продемонстровано, як обраховується вихід згорткового шару.

2.1.1.2 Використання багатьох фільтрів

В приведеному прикладі в попередньому пункті зображено роботу мережі, в якій два фільтри, проте фільтрів може бути набагато більше. Стандартно розмірність фільтрів являє собою степінь числа 2, наприклад 16, 32, 64.

Наприклад, якщо використати 32 фільтри, то ми отримаємо карту активацій розміром 30x30x32 (Рисунок 2.8).

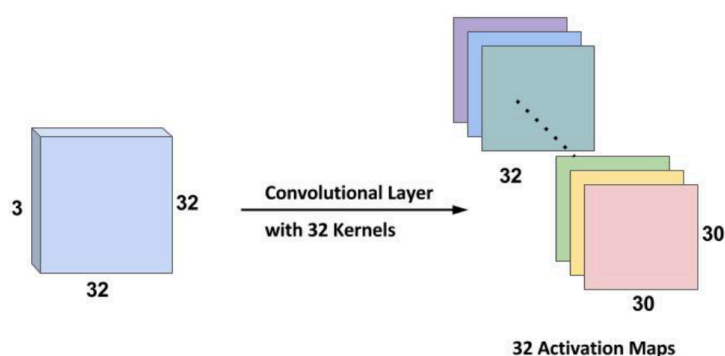


Рисунок 2.8 – Карта активацій з 32 шарами

Так як під час роботи згорткових шарів розмірність зображень постійно зменшується, то для коректної роботи математичних операцій необхідно, щоб розмірності вхідних і вихідних матриць залишались однаковими. Для цього використовують процедуру заповнення нулями.

2.1.1.3 Заповнення нулями

Кожний згортковий шар зменшує розмірність зображення (наприклад з 32 до 30). Для зручності, на практиці прийнято приводити вихідну розмірність до вхідної, заповнюючи необхідну кількість пікселів нулями. Так додаючи по 1 ряду

нулів з кожної сторони зображення 30x30 ми отримаємо зображення 32x32. Для обрахунку необхідного розміру доповнення використовується формула:

$$M = \frac{N-F+2P}{S} + 1 \quad (2.2)$$

де N – вхідна розмірність, F – розмірність фільтра, S – крок, M – вихідна розмірність. Рівняння розв’язується для P , його значення і буде значенням необхідного доповнення.

2.1.3 Об’єднуючі шари

Зазвичай в сучасних CNN використовуються не тільки згорткові, а й об’єднуючі шари.

Ціллю об’єднуючих шарів є зменшення просторового розширення карт ознак, і таким чином досягнення просторової інваріантності до спотворень на вході [16,18], а також зменшення кількості необхідних обчислень. Спочатку загальноприйнятою практикою було використання середніх значень в шарах об’єднання [16]. В нових моделях [19,20] об’єднуючі шари передають не середнє, а максимальне значення в межах кожного рецептивного поля на наступний шар. Формально, це вибір найбільшого елемента в кожному рецептивному полі, і може бути описано наступною формулою:

$$Y_{kij} = \max_{(p,q) \in \mathcal{R}_{ij}} x_{kpq} \quad (2.3)$$

де результат операції об’єднання, зв’язаний з k -ою картою ознак, визначається як Y_{kij} . x_{kpq} позначає елемент (p, q) , який міститься в області об’єднання \mathcal{R}_{ij} , яка відображає рецептивне поле навколо положення (i, j) [18].

На рисунку 2.9 зображено різницю між підходами об'єднання з використанням середнього і максимального значень

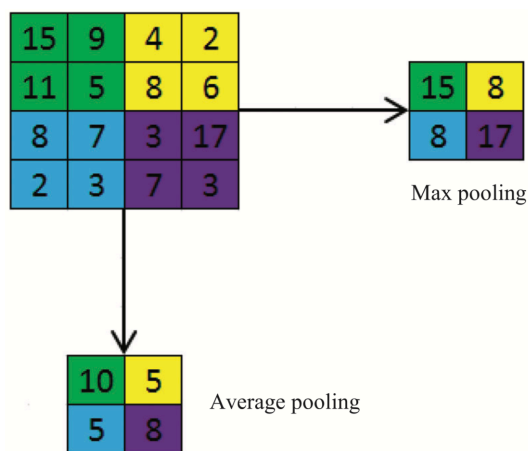


Рисунок 2.9 Підходи до об'єднання

На практиці зазвичай використовують процес об'єднання з фільтром розмірністю 2, це дозволяє зменшити розмірність зображення в 2 рази.

2.1.4 Повнозв'язні шари

Згорткові та об'єднуючі шари являють собою тіло самої CNN, коли ж повнозв'язні шари необхідні для класифікації об'єктів. Повнозв'язний шар працює з векторами, на відміну від згорткових, тому вихід згорткової частини мережі з просторової матриці перетворюється у один вектор.

Повнозв'язні шари йдуть в кінці мережі і інтерпретують ознаки, отримані раніше в певні класи і виконують функції по прийняттю рішень [21]. Для проблем класифікації стандартно використовується оператор softmax [18 - 21]. Проте на початках використовували класифікатор на базі радіально-базисних функцій [15].

Повнозв'язний шар звертається до виходу попереднього шару і визначає ознаки, які найбільш зв'язані з певним класом. Якщо мережа вважає, що на

зображенні кіт, то у високорівневих карт ознак, які відповідають за ознаки кота, наприклад 4 лапи, 2 вуха, хвіст, тощо, будуть високі значення. Повнозв'язний шар оцінює те, що високорівневі функції пов'язані з певним класом і мають певні ваги, так при розрахунку добутку ваг з вагами попереднього шару отримуються правильні ймовірності приналежності для різних класів.

2.1.5 Аналіз структури вхідних зображень

Описавши всі шари мережі, можна перейти до процесу навчання. Цей процес дозволяє мережі навчитись розрізняти різні вхідні зображення і відносити їх до класів. Навчання відбувається на певних вибірках зображень. CNN сприймають зображення як тривимірні об'єкти, а не площини, які вимірюються тільки по ширині і висоті. Це відбувається тому, що цифрові зображення мають кольірну схему RGB (червоний, зелений, синій), а мережа сприймає кожне зображення як три шари кольорів.

Таким чином CNN отримує на вхід прямокутне зображення, висота і ширина якого вимірюються в пікселях і глибиною в три шари – дані шари називаються каналами і відповідають кожному з кольорів схеми RGB.

2.1.6 Поняття тензора

Враховуючи специфіку вхідних зображень в CNN, далі з ними потрібно проводити математичні операції. Щоб описати тривимірні зображення, з якими проводяться операції в CNN існує поняття тензора.

Тензори - це матриці чисел з додатковими вимірами.

Тензори важко візуалізувати, тому поняття тензора простіше пояснити по аналогії. Скаляр - це число, наприклад, 7; вектор – це декілька чисел (наприклад, [7,8,9]); матриця - це прямокутна сітка з чисел, що займає кілька рядків і стовпчиків. Геометрично, якщо скаляр є нульовимірною точкою, то вектор - це одновимірна лінія, матриця - це двовимірна площина, стек матриць - це тривимірний куб, а коли кожен елемент цих матриць має стек карт з особливостями, зв'язаних з нею, вводять четвертий вимір.

Наприклад, 2x2 матриця:

$$\begin{bmatrix} 1 & 2 \\ 5 & 8 \end{bmatrix}$$

Тензор охоплює виміри поза цією 2D площиною. Ви можете легко уявити тривимірний тензор, за допомогою масиву чисел розташованих у кубі. Наприклад 2x3x2 тензор, представлений в одній площині:

$$\begin{bmatrix} \begin{pmatrix} 2 \\ 3 \end{pmatrix} & \begin{pmatrix} 3 \\ 5 \end{pmatrix} & \begin{pmatrix} 4 \\ 7 \end{pmatrix} \\ \begin{pmatrix} 3 \\ 4 \end{pmatrix} & \begin{pmatrix} 4 \\ 6 \end{pmatrix} & \begin{pmatrix} 5 \\ 8 \end{pmatrix} \end{bmatrix}$$

Візуальне представлення тензора зображено на рисунку 2.11.

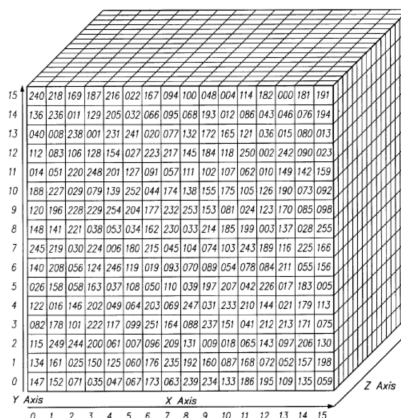


Рисунок 2.11 – Візуальне представлення тензора

2.1.7 Процес навчання

Для того, щоб у CNN появилась здатність класифікувати зображення, мережу необхідно навчити це робити. Під час навчання мережа запам'ятовує, те, якими параметрами описуються які класи, що в подальшому дозволяє мережі, отримавши ознаки зображення, розпізнати до якого класу вони належать.

CNN використовують алгоритми навчання для настройки ваг. Найпоширенішим алгоритмом являється *backpropagation* [15, 22]. Цей алгоритм обраховує похибку, щоб визначити, як потрібно змінити ваги, щоб мінімізувати помилки і максимізувати ефективність.

Метод зворотнього поширення помилки (*backpropagation*) можна розділити на 4 окремих блоки:

- Пряме поширення.
- Функція втрат.
- Зворотнє поширення.
- Коригування ваг.

Під час прямого поширення береться тренувальне зображення, і пропускається через всю мережу. Ваги ініціалізуються випадковим чином (наприклад одиницями), тому мережа не зможе адекватно визначити клас зображення. Так з'являється функція втрат (похибка) – вона може бути різною, проте зазвичай використовується середньоквадратичне відхилення:

$$E_{total} = \sum \frac{1}{2} (target - output)^2 \quad (2.4)$$

Для перших зображень дана похибка буде дуже великою. Похибку необхідно мінімізувати, для цього визначаються ваги, які максимально впливали на похибку. Для мінімізації використовують різні методи математичної оптимізації. На практиці зазвичай використовують методи градієнтного спуску.

Дані методи базуються на пошуку градієнту цільової функції і зміні ваг таким чином, щоб вони рухались в сторону градієнту:

$$w = w_i - \eta \frac{dL}{dw} \quad (2.5)$$

де w – вага, w_i – початкова вага, η – швидкість навчання.

Швидкість навчання – це параметр, який задається експертом і який впливає на крок зміни ваг. Висока швидкість навчання приводить до швидшого навчання мережі, проте в той же ж час понижує точність, так як через великі кроки може не бути досягнуто прийнятного мінімуму (Рисунок 2.10).

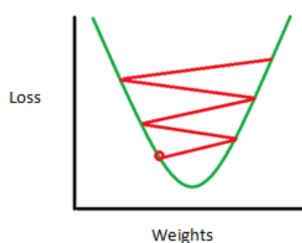


Рисунок 2.10 – Проскакування мінімуму

Процес проходження всіх 4 блоків називається епохою. Кількість епох для кожного зображення визначається експертно і напряму впливає на швидкість і точність навчання мережі.

Процес навчання є останнім етапом в побудові класифікатора. Після завершення навчання класифікатор може використовуватись на практиці.

Описавши всі елементи в CNN можна перейти до процесу побудови класифікатора, який б ефективно виконував задачу класифікації товарів.

2.2 Аналіз запропонованої структури нейронної мережі для класифікації товарів в ритейлі

Для проведення ефективної класифікації товарів необхідно побудувати CNN, яка б дозволила максимально точно класифікувати специфічну вибірку зображень. В поставленій задачі цією вибіркою є товари на полицях в торгових точках.

Для вибору структури мережі було проаналізовано багато різних підходів до розпізнавання, запропонованих різними дослідниками в останні 8 років, та проведено велику кількість тестів на розпізнавання з використанням існуючих структур нейронних мереж. Проте жодний із тестів не показав задовільних результатів, тому було прийняте рішення розробити нову структуру мережі, яка буде максимально ефективно працювати на специфічних даних з сфери ритейлу.

В пунктах 3.2 і 4.2.2 детально описано процес вибору, а також аргументовано доцільність використання запропонованої архітектури CNN.

Запропонована структура CNN зображена на рисунку 2.12.

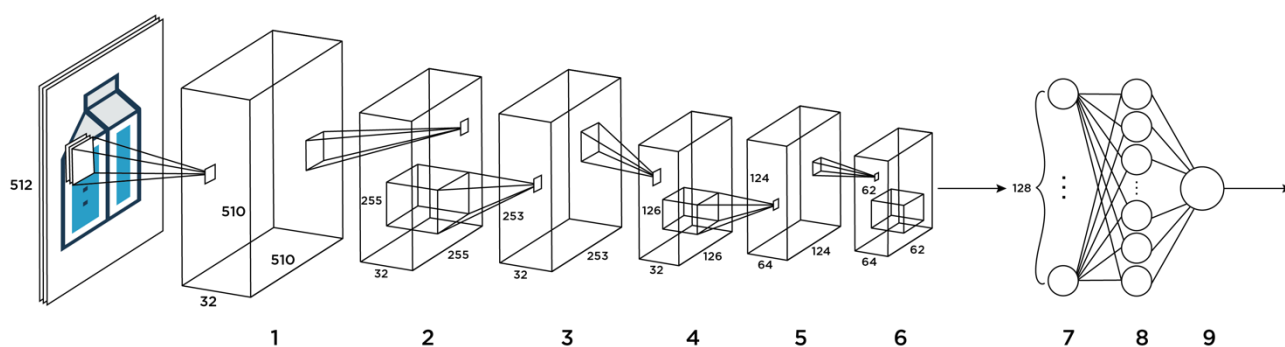


Рисунок 2.12 – Структура запропонованого класифікатора

В пунктах 2.3 – 2.9 детально описано структуру запропонованого класифікатора, процес підготовки даних та навчання.

2.3 Опис шарів запропонованої CNN

Запропонована мережа має послідовну структуру з трьома шарами згортки, трьома шарами об'єднання та двома повнозв'язними шарами (Рисунок 2.12).

1. Перший шар в мережі – згортковий з 32 картами ознак. На вхід отримує зображення розміру $512 \times 512 \times 3$. На виході отримується зображення розміром $510 \times 510 \times 32$. Розмір фільтру 3×3 , крок 1. Функція активації – нелінійна ReLU (детально описана в пункті 2.9).
2. Другий шар в мережі – об'єднувальний. Ядро розміру 2×2 . Вибирає максимальний елемент з області 2×2 . На вхід отримує вихід попереднього шару. На виході розмірність зображення зменшується в 2 рази і стає $255 \times 255 \times 32$.
3. Третій шар в мережі - згортковий з 32 картами ознак. Розмір фільтру 3×3 , крок 1, активація ReLU. На виході отримуємо зображення розміром $253 \times 253 \times 32$.
4. Четвертий шар в мережі – об'єднувальний. Даний шар аналогічний до другого. На виході розмірність зображення $126 \times 126 \times 32$.
5. П'ятий шар в мережі – згортковий з 64 картами ознак. Розмір фільтру 3×3 , крок 1, активація ReLU. Розмірність виходу $124 \times 124 \times 64$.
6. Шостий шар в мережі – об'єднувальний. Даний шар аналогічний до другого. Розмірність вихідного зображення $62 \times 62 \times 64$.
7. Сьомий шар проводить процес перетворення згорткового виходу в один довгий вектор, для того, щоб його можливо було подати на вхід повнозв'язного шару.
8. Сьомий шар в мережі – повнозв'язний з 128 нейронами, функцією активації ReLU. До цього шару застосовується допоміжна функція – dropout (детально описаний в пункті 2.10).

9. Восьмий шар в мережі – повнозв'язний і являє собою вихід. Він складається з одного нейрона, функція активації – сигмоїд (детально описана в пункті 2.9), дана функція активації ідеально підходить для задачі бінарної класифікації.

2.4 Аналіз використаних функцій активації

Вибір функції активації впливає на час навчання мережі, що суттєво впливає на продуктивність великих мереж на великих наборах даних [19].

2.4.1 Нелінійна функція ReLU (Rectified linear units)

ReLU була впроваджена в 2010 році Нейром і Хінтоном для глибоких машин Больцмана. Широкого поширення вона набула в 2012 році після її використання Крижевським в AlexNet. Рисунок 2.13 порівнює час навчання з використанням ReLU (суцільна лінія) і з \tanh (гіперболічним тангенсом – пунктирна лінія) для чотирьохшарової згорткової мережі [19], навченої на наборі CIFAR-10. Мережа, в якій використовували ReLU навчалась в 10 разів швидше.

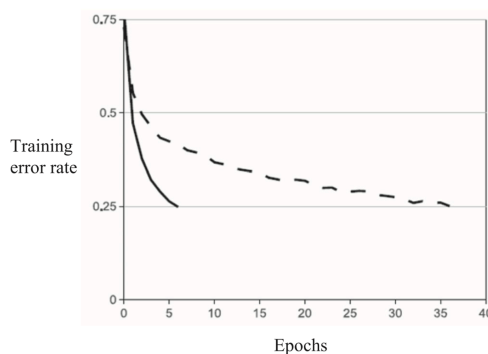


Рисунок 2.13 – Час навчання ReLU і \tanh

ReLU являє собою кусково-лінійну функцію, спрощена форма якої виглядає наступним чином:

$$f(x) = \max(x, 0) \quad (2.6)$$

Графічно ReLU зображено на рисунку 2.14.

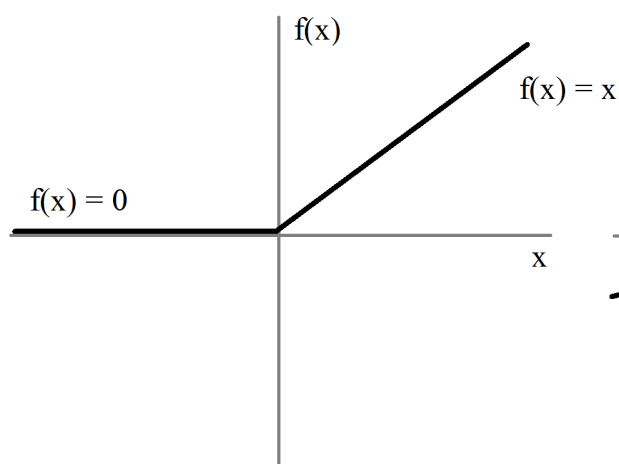


Рисунок 2.14 – Функція ReLU

ReLU зберігає тільки додатну частину активації, від’ємна ж перетворюється в нуль, а інтегрований оператор максимуму прискорює обчислення. Використання ReLU не змінює розмірності шару. При цьому збільшує нелінійність як функції прийняття рішення так і всієї моделі, ніяк не впливаючи на рецепторну частину згорткового шару.

Робота ReLU на реальних даних зображена на рисунку 2.15.

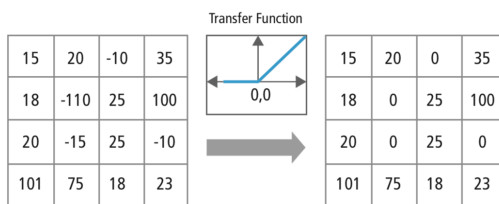


Рисунок 2.15 – Приклад роботи ReLU

2.4.2 Функція активації сигмоїд

Сигмоїд являє собою одну із найпростіших функцій активації, проте ідеально підходить для специфічної задачі. В поставленій задачі відсутня категоріальна класифікація. Значення виходу потрібне тільки 1 або 0 (чи наближені до них значення). Особливістю сигмоїда є його існування тільки між 0 та 1. Тому він використовується для моделей, в яких є вибір між двома об'єктами, присутня бінарна класифікація. Функція являється диференційованою і монотонною. Сигмоїд представлений наступною формулою:

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1} \quad (2.7)$$

Сигмоїд є S подібною функцією. Графічно сигмоїд представлений на рисунку 2.16.

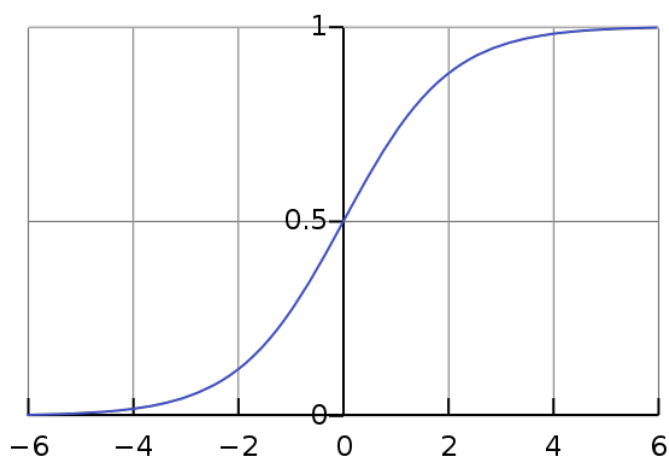


Рисунок 2.16 – Графік функції сигмоїд

Попри те, що сигмоїд розмиває градієнт, має повільну збіжність і не використовується в новітніх мережах для задачі бінарної класифікації його використання є доцільним.

2.5 Аналіз функції Dropout

Для того, щоб класифікатор був ефективний не тільки на навчальній та валідаційних вибірках, необхідно використовувати допоміжні засоби, які дозволять зменшити прив'язку ваг класифікатора до специфічних наборів зображень. Для такої задачі використовують механізми регуляризації. В запропонованій мережі використовується Dropout.

В моделях з великою кількістю зв'язків дуже часто виникає проблема перенавчання – коли модель вдало працює виключно на тренувальних даних. Найпростішим методом боротьби з перенавчанням є збільшення кількості даних, проте це приводить до суттєво збільшення необхідних обчислювальних ресурсів [20]. Dropout являється найефективнішим засобом для вирішення проблеми перенавчання.

При використанні Dropout нейрони виключаються з шару з ймовірністю p , а також залишаються з ймовірністю $p - 1$.

Ймовірність виключення кожного нейрону однакова, це означає наступне:
За умов:

- а) $h(x) = xW + b$ – лінійна проекція вхідного d_i – мірного вектора x на d_h –мірний простір вихідних значень;
- б) $a(h)$ – функція активації.

використання Dropout до даної проекції в процесі навчання можна представити як змінену функцію активації:

$$f(h) = D \odot a(h) \quad (2.8)$$

де $D = (X_1, \dots, X_{d_h})$ – d_h мірний вектор випадкових величин X_i , розподілених по закону Бернуллі.

Розглянемо застосування Dropout до i -го нейрона, де $P(X_i = 0) = p$. Так як на етапі навчання нейрон залишається в мережі з ймовірністю q , на етапі тестування необхідно емулювати поведінку ансамблю мереж, які

використовувались при навчанні. Для цього на етапі тестування функцію активації треба помножити на коефіцієнт q , тоді:

На етапі навчання Dropout використовується:

$$O_i = X_i a(\sum_{k=1}^{d_i} w_k x_k + b) \quad (2.9)$$

На етапі тестування Dropout використовується:

$$O_i = qa(\sum_{k=1}^{d_i} w_k x_k + b) \quad (2.10)$$

Графічно Dropout представлений на рисунку 2.17.

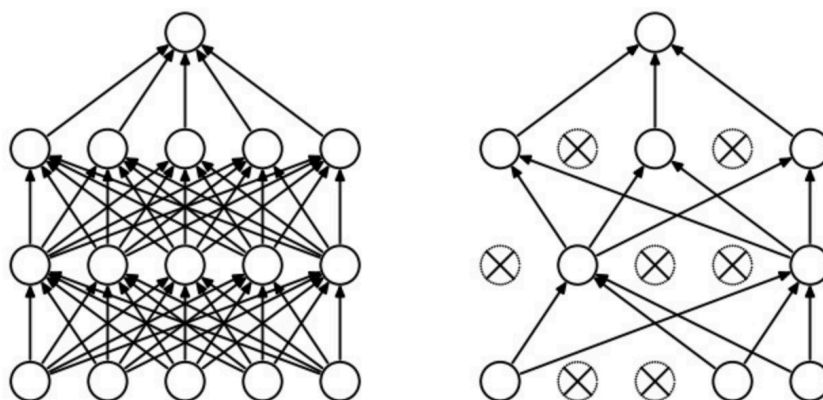


Рисунок 2.17 – Приклад використання техніки Dropout

2.6 Аналіз функції ціни (похибки) для задачі бінарної класифікації

Одним з найважливіших етапів в побудові класифікатора являється вибір правильної функції ціни (похибки). Ця функція використовується для визначення подібності між виходом і бажаним виходом класифікатора. Саме

функція ціни піддається оптимізації для налаштування максимальної подібності між виходом класифікатора і реальними класами.

Для задачі бінарної класифікації за допомогою згорткових нейронних мереж оптимальним рішенням є використання функції бінарної кросс-ентропії. Так як вихід мережі може приймати тільки значення між 0 та 1. Немає необхідності використовувати функції, які дозволять визначати приналежність до багатьох класів. Тому бінарна кросс-ентропія ідеально підходить для вирішення поставленої задачі.

Для дискретних значень p і q значення кросс-ентропії вираховується як:

$$H(p, q) = - \sum_x p(x) \log(q(x)) \quad (2.11)$$

Бінарна кросс-ентропія, або логарифмічна помилка базується на логістичній регресії, яка дозволяє визначити ймовірності приналежності до двох визначених класів, а саме 0 і 1. Модель логістичної регресії прогнозує вихід $y = \{0,1\}$ при заданому вході x . Ймовірність моделюється з використанням логістичної функції:

$$g(z) = 1/(1 + e^{-z}) \quad (2.12)$$

Ймовірність отримати вихід $y = 1$ можна описати наступним чином:

$$q_{y=1} = \hat{y} \equiv g(w \cdot x) = 1/(1 + e^{-w \cdot x}) \quad (2.13)$$

де вектор ваг w оптимізується за допомогою запропонованого алгоритму оптимізації. З іншої сторони ймовірність отримати вихід $y = 0$ можна описати наступним чином:

$$q_{y=0} = 1 - \hat{y} \quad (2.14)$$

Реальні ймовірності (які отримуються мережею) описуються наступним чином:

$$p_{y=1} = y \quad (2.15)$$

$$p_{y=0} = 1 - y \quad (2.16)$$

Поставивши $p \in \{y, 1 - y\}$ і $q \in \{y', 1 - y'\}$ ми можемо використати кросс-ентропію для визначення неподібності між p і q :

$$H(p, q) = -\sum_i p_i \log(q_i) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (2.17)$$

Описавши ідею використання кросс-ентропії можна перейти до визначення функції ціни кросс-ентропії. Припустимо потрібно навчити нейрон з трьома входами x_1, x_2, x_3 і вагами w_1, w_2, w_3 і зміщення b (Рисунок 2.18):

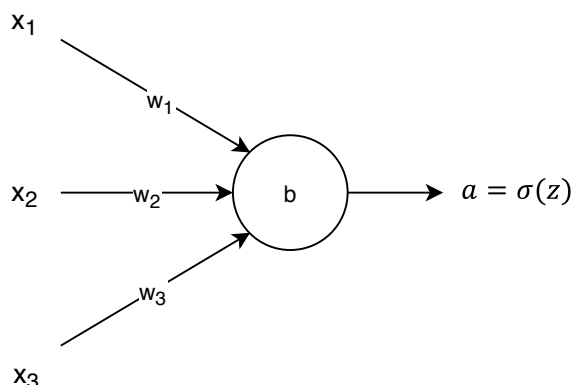


Рисунок 2.18 – Приклад нейрона

Виходом нейрона буде $a = \sigma(z)$, де $z = \sum_j w_j x_j + b$ являє собою зважену суму всіх входів. Тоді для такого нейрона можна визначити функцію ціни з використанням бінарної кросс-ентропії:

$$C = -\frac{1}{n} \sum_x [y \log a + (1 - y) \log(1 - a)] \quad (2.18)$$

де n – загальна кількість елементів в тренувальній вибірці, x – кількість входів, y – бажаний вихід.

Далеко не очевидно, що використання такої функції вирішить проблему уповільнення навчання, і взагалі, що таку функцію можна використовувати як функцію ціни.

Два наступні твердження аргументують можливість використання кросс-ентропії як функції ціни:

- а) Значення C завжди невід'ємне. Всі елементи в сумі є від'ємними, так як обидва логарифми отримують значення в діапазоні від 0 до 1, а також є знак – перед сумою.
- б) Якщо значення бажаного виходу буде близьким до реального для всіх тренувальних входів x , то значення кросс-ентропії буде близьке до 0. Таким чином низьке значення C вказуватиме на подібність реального і бажаного виходів.

Головною перевагою використання кросс-ентропії є відсутність проблеми з уповільненням навчання. Щоб це продемонструвати обрахуємо часткове відхилення цінової функції кросс-ентропії застосовуючи ланцюгове правило двічі

$$\begin{aligned}\frac{dC}{dw_j} &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)} \right) \frac{d\sigma}{dw_j} = \\ &= -\frac{1}{n} \sum_x \left(\frac{y}{\sigma(z)} - \frac{1-y}{1-\sigma(z)} \right) \sigma'(z) x_j\end{aligned}\tag{2.19}$$

Після спрощень це виглядатиме наступним чином:

$$\frac{dC}{dw_j} = \frac{1}{n} \sum_x \frac{\sigma'(z) x_j}{\sigma(z)(1-\sigma(z))} (\sigma(z) - y)\tag{2.20}$$

Використавши, що:

$$\sigma(z) = 1/(1 + e^{-z})\tag{2.21}$$

А також, що:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z)) \quad (2.22)$$

Ми можемо ще спростити вираз:

$$\frac{dC}{dw_j} = \frac{1}{n} \sum_x x_j (\sigma(z) - y) \quad (2.23)$$

З даної формули чітко видно, що швидкість навчання ваг нейрона контролюється виразом $\sigma(z) - y$, тому чим більша помилка, тим швидше нейрон навчається. Це саме те, що очікувалось – навчання не уповільнюється через використання $\sigma'(z)$, на відміну від стандартних квадратичних помилок.

2.7 Аналіз запропонованого модифікованого адаптивного методу стохастичного градієнтного спуску

Після того, як було обрано функцію ціни необхідно обрати метод оптимізації. Оптимізатор переналаштовує ваги мережі таким чином, щоб з кожною новою ітерацією функція ціни зменшувалась. Задачею оптимізації є приведення функції ціни до певного близьконулевого значення, що в свою чергу означатиме, що класифікатор ефективно визначає клас вхідного зображення. Від вибору методу оптимізації залежить не тільки точність навчання мережі, а й ефективність. Якщо вибрано неправильний оптимізатор, то функція ціни може ніколи не досягнути мінімуму, коли при виборі ефективного оптимізатора мінімум досягається за мінімально можливою кількістю кроків.

Сьогодні в машинному навчанні загальноприйнятим є використання методів оптимізації, що базуються на стохастичних градієнтних методах, так як,

якщо функція ціни є диференційована, то градієнтний спуск в будь-якому випадку приведе до мінімуму, ключове питання, наскільки це буде ефективно. Цільові функції зазвичай є стохастичними, в такому випадку найпоширенішим методом оптимізації являється SGD, метод стохастичного градієнтного спуску [19,23].

Запропонований алгоритм базується на SGD, потребує тільки градієнти першого порядку і використовує мало обчислювальних ресурсів. Метод обраховує індивідуальні адаптивні швидкості навчання для різних параметрів за рахунок оцінок експоненційного ковзного середнього градієнту і квадрату градієнту. Метод комбінує переваги двох відомих методів: AdaGrad [24] і RMSProp [25]. Основними перевагами являється те, що оновлення параметрів є інваріантні до масштабування градієнта, розмір його кроку є обмежений гіперпараметром кроку, метод не потребує стаціонарної цілі, працює з розрідженими градієнтами, а також виконує отжиг для виходу і поглиблення в локальні мінімуми.

2.7.1 Алгоритм

Обов'язкові вхідні дані:

α : крок;

$\beta_1, \beta_2 \in [0,1)$: експоненційні швидкості розкладу;

$f(\theta)$: стохастична цільова функція з параметром θ ;

θ_0 : Початковий вектор параметрів;

Псевдокод:

$\mu_0 \leftarrow 0$ (ініціалізація вектора експоненційного ковзного середнього градієнта);

$\sigma_0 \leftarrow 0$ (ініціалізація вектора експоненційного ковзного середнього квадрата градієнта);

$t \leftarrow 0$ (ініціалізація моменту часу);

while θ_t : not converged **do**

$t \leftarrow t + 1$;

$g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (обрахунок градієнтів відносно стохастичної цілі в момент t);

$\mu_t \leftarrow \beta_1 \cdot \mu_{t-1} + (1 - \beta_1) \cdot g_t$ (корекція експоненційного ковзного середнього градієнта (оцінка математичного сподівання));

$\sigma_t \leftarrow \beta_2 \cdot \sigma_{t-1} + (1 - \beta_2) \cdot g_t^2$ (корекція експоненційного ковзного середнього квадрата градієнта (оцінка дисперсії));

$\hat{\mu}_t \leftarrow \mu_t / (1 - \beta_1^t)$ (обрахунок скорегованого мат сподівання);

$\hat{\sigma}_t \leftarrow \sigma_t / (1 - \beta_2^t)$ (обрахунок скорегованої дисперсії);

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{\mu}_t / (\sqrt{\hat{\sigma}_t} + \varepsilon)$ (оновлення параметрів);

end while

return θ_t ;

g_t^2 означає поелементний добуток $g_t \odot g_t$.

2.7.2 Правило оновлення кроку

Ефективний крок в момент часу t визначається як:

$$\Delta_t = \alpha \cdot \hat{\mu}_t / \sqrt{\hat{\sigma}_t} \quad (2.24)$$

Ефективний крок має дві верхні границі:

$$|\Delta_t| \leq \alpha \cdot \frac{(1 - \beta_1)}{\sqrt{1 - \beta_2}}, \text{ якщо } (1 - \beta_1) > \sqrt{1 - \beta_2} \quad (2.25)$$

$$|\Delta_t| \leq \alpha, \text{ в інших випадках}$$

Перший варіант буває тільки у випадку, якщо градієнт був рівний нулю на всіх кроках окрім поточного.

Так величина ефективного кроку фактично обмежена гіперпараметром α . Це дозволяє експертно вирахувати таке значення параметра α при якому оптимальне значення буде знайдене за певну невелику кількість ітерацій. Другий елемент, який визначає ефективний крок є відношення $\hat{\mu}_t/\sqrt{\hat{\sigma}_t}$. Чим менше це значення, тим ближчим буде ефективний крок до нуля. Це бажана ситуація, так як зі зменшенням значення $\hat{\mu}_t/\sqrt{\hat{\sigma}_t}$, збільшується невизначеність чи відповідає напрямок $\hat{\mu}_t$ напрямку істинного градієнта. Ближче до оптимуму значення $\hat{\mu}_t/\sqrt{\hat{\sigma}_t}$ стає близьким до нуля, що веде до зменшення розміру кроку, що являє собою автоматичний отжиг. $|\Delta_t|$ інваріантний до масштабу градієнта. Так якщо помножити градієнт на константу c , то $\hat{\mu}_t$ і $\hat{\sigma}_t$ помножаться на c і c^2 відповідно, що приведе до інваріантності:

$$(c \cdot \hat{\mu}_t)/(\sqrt{c^2 \cdot \hat{\sigma}_t}) = \hat{\mu}_t/\sqrt{\hat{\sigma}_t} \quad (2.26)$$

2.7.3 Корекція μ_t і σ_t

Опишемо складніший процес корекції дисперсії, для математичного сподівання він аналогічний.

σ_t коригується за наступною формулою:

$$\sigma_t = \beta_2 \cdot \sigma_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (2.27)$$

Ця формула може бути переписана як функція градієнтів на всіх попередніх моментах часу:

$$\sigma_t = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \cdot g_i^2 \quad (2.28)$$

Потрібно дізнатись, як очікуване значення дисперсії $\mathbb{E}[\sigma_t]$ відноситься до реального значення $\mathbb{E}[g_i^2]$, щоб можна було скорегувати невідповідність між ними.

$$\begin{aligned} \mathbb{E}[\sigma_t] &= \mathbb{E} \left[(1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \cdot g_i^2 \right] = \\ &= \mathbb{E}[g_t^2] \cdot (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} + \zeta = \\ &= \mathbb{E}[g_t^2] \cdot (1 - \beta_2^t) + \zeta \end{aligned} \quad (2.29)$$

де $\zeta = 0$, якщо g_i^2 стаціонарний, або взято малим в іншому випадку. $(1 - \beta_2^t)$ включає в себе гіперпараметр, який задається експертом і використовується для коригування невідповідності [26].

2.8 Процес підготовки даних для навчання запропонованої мережі

Після завершення вибору структури класифікатора, цінової функції та оптимізатора необхідно провести навчання класифікатор на необхідних даних. Так як дані, які використовуються є специфічними і доступ до них обмежений, було запропоновано наступний процес підготовки вибірок зображень, який можна розділити на такі етапи:

1. Підготовка початкового набору фотографій і розподіл фотографій по класах (у випадку роботи з запропонованою структурою НМ

використовуватимуться 2 класи, а саме клас “об’єкт” і клас “не об’єкт”).

2. Розбиття набору фотографій кожного з класів на навчальну, валідаційну та тестову вибірки.
3. Урізноманітнення зображень за допомогою функцій деформації.
4. Приведення зображень до необхідного розміру.

2.8.1 Підготовка початкового набору зображень

Для кожного класифікатора необхідно підготувати вибірку з 2000 фотографій. 1000 фотографій, на яких зображено об’єкт і 1000 фотографій без об’єкту, з них 500 фотографій на яких зображено різноманітні об’єкти, що зустрічаються в ритейлі і 500 фотографій з зображеннями товарів, які є аналогічні об’єкту (наприклад для позиції “Молоко Селянське” це можуть бути інші види молока).

Для кожного об’єкту готується унікальний набір фотографій.

1. Фотографії з об’єктом.

Вибірка фотографій, на яких зображений об’єкт отримується шляхом проведення відеозйомки об’єкту таким чином, щоб кожен з його сторін було знято і при цьому об’єкт знаходився по центру. Наступним кроком відео обробляється спеціально розробленою програмою, яка витягує з відео 1000 зображень, які ефективно описують об’єкт. Програма автоматично надає кожній фотографії мітку класу “об’єкт”.

2. Фотографії без об’єкту.

Перші 500 фотографій це набір різноманітних об’єктів, які зустрічаються в сфері ритейлу. Наступні 500 фотографій – це фотографії об’єктів аналогів. Якщо аналогів 5, то 500 фотографій будуть розподілені по 100 для кожного з

аналогів, якщо 10, то відповідно по 50 фотографій для кожного з аналогів. Даліше отриманій 1000 фотографій надається мітка класу “не об’єкт”.

2.8.2 Розбиття фотографій на навчальну, валідаційну і тестову вибірки

Процес розбиття відбувається з використанням рандомізатора. 70 відсотків випадково вибраних фотографій визначаються як навчальна вибірка, 15 відсотків як валідаційна і 15 як тестова для всіх класифікаторів.

2.8.3 Урізноманітнення зображень за допомогою функцій деформації

Так як початкова вибірка зображень є дуже малою і не може ефективно описати всі можливі зміни в освітленні, розміщенні та можливі оптичні деформації об’єкту, необхідно розширити вибірку за допомогою уже існуючих зображень, деформуючи їх.

З фотографіями проводяться наступні деформації на програмному рівні:

- зміна освітленості зображення.
- повороти.
- дзеркальне відображення.
- деформації кольору.
- деформації масштабу.

Нові фотографії, отримані шляхом деформацій додаються до вибірки. Таким чином вибірка розширюється і дозволяє повністю описати об’єкт.

2.8.4 Приведення зображень до необхідного розміру

На вхід класифікатора подаються зображення певного розміру. Для запропонованої структури це квадратні зображення розміром 512 на 512 пікселів. Це великі зображення, більшість відомих згорткових мереж опрацьовують суттєво менші зображення, проте для специфічної задачі в ритейлі необхідно розпізнавати невеликі надписи на товарах, і це змушує використовувати великі зображення. Початкові зображення мають неквадратний формат і різні розміри.

Система автоматично приводить їх до необхідного розміру процесом деформування. Це призводить до візуальної деформації зображення, але не впливає на ефективність навчання класифікатора.

2.9 Аналіз процесу навчання і тестування запропонованої мережі

Даний етап є найважливіший і фінальний в роботі з класифікатором. Від нього залежить чи побудований класифікатор зможе ефективно виконувати поставлену задачу класифікації.

На цьому етапі відбувається процес налаштування всіх основних параметрів мережі.

Для того, щоб навчання мережі було ефективним необхідно визначитись з головними параметрами, які б дозволили вирішити поставлену задачу. Вибір параметрів проводиться експертом і повністю залежить від його аналітичних і практичних знань.

В попередніх пунктах описано основні параметри, які було обрано для вирішення задачі класифікації, а саме:

- структуру нейронної мережі;
- функції активації;

- метод регуляризації;
- функцію ціни (похибку);
- метод оптимізації.

Підготувавши дані, необхідні для навчання можна приступити безпосередньо до процесу навчання. Для роботи з тензорами використовується середовище Tensorflow, яке дозволяє з невеликими затратами проводити операції на великих об'ємах даних. Процес навчання є стандартним. На вхід мережі послідовно подаються фотографії кожного з класів. Мережа витягує з кожної фотографії особливі властивості і настраює ваги. Процес зупиняється по закінченню певної кількості епох. Для вирішення поставленої задачі достатньо використовувати 30 епох, це дозволяє отримати достатньо точний результат.

2.10 Використання запропонованого класифікатора

Після завершення етапу навчання і досягнення задовільних результатів точності класифікатор може використовуватись для вирішення задачі класифікації товарів.

Для кожного унікального товару під час навчання зберігаються ваги, при яких класифікатора зміг максимально точно класифікувати об'єкт. Ваги зберігаються в хмарному сховищі у вигляді кешу і вивантажуються індивідуально для кожного товару. Для класифікації товару, на вхід мережі подається зображення, на якому знаходиться некласифікований об'єкт. Зображення обробляється системою сегментації таким чином, що на ньому знаходиться тільки один товар.

Після цього, вхідне зображення, пропускається через класифікатор, який завантажено з вагами, які відповідають товару, який мав би знаходитись на зображенні. На виході отримується ймовірнісне значення приналежності товару на зображенні до одного з класів – “об'єкт” і “не об'єкт”. Якщо товар відповідає

тому, який має бути на планогамі, тоді система його віднесе до класу об'єкт, якщо ж ні, то система вкаже, що на зображенні не бажаний товар і викладка не відповідає планогамі.

Висновки за розділом 2

У даному розділі було розглянуто та описано основний підхід до класифікації зображень – глибокі згорткові нейронні мережі, на яких базується запропонований рішення: глибока згорткова нейронна мережа з індивідуальною архітектурою і модифікованими адаптивним методом стохастичного градієнтного спуску.

Досліджено стандартні структури згорткових нейронних мереж. Описано їх елементи та способи їх застосування.

Досліджено і описано функції активації, які використовуються в роботі.

Досліджено і описано метод регуляризації Dropout і наведено приклад його використання.

Описано процес підготовки даних, який являється специфічним для поставленої задачі.

Запропоновано унікальну структуру глибокої згорткової нейронної мережі для задачі класифікації товарів.

Запропоновано модифікований адаптивний метод стохастичного градієнтного спуску, який дозволив в рази прискорити процес навчання класифікатора, а також підвищити його точність.

Запропоновано спосіб використання створеного класифікатора для задачі класифікації зображень в сфері ритейлу.

РОЗДІЛ 3. АНАЛІЗ АРХІТЕКТУРИ ПРОГРАМНОГО ПРОДУКТУ

3.1 Обґрунтування вибору платформи та мови програмування

Інтелектуальна система мерчендайзингу представляє собою набір окремих модулів (підсистем), які між собою пов'язані тим чи іншим завданням. Такими модулями є:

1. модуль управління камерами в ТТ, зчитування зображень в реальному часі;
2. модуль роботи з БД;
3. модуль сегментації зображень;
4. модуль класифікації товарів;
5. модуль аналізу результатів;
6. модуль взаємодії з персоналом.

Оскільки наявна велика кількість окремих модулів, то в якості мови програмування доцільно обрати Python.

Python є інтерпретуємою мовою програмування. Python є надзвичайно простий і містить невелику кількість ключових слів, в той же час будучи дуже гнучким. Це мова більш високого рівня ніж Pascal, C ++ і, звичайно C, що досягається, в основному, за рахунок вбудованих високорівневих структур даних (списки, словники, тюпи).

Беззаперечною перевагою є те, що інтерпретатор Python реалізований практично на всіх платформах і операційних системах. Першою такою мовою була C, проте її типи даних на різних машинах могли займати різну кількість пам'яті і це служило деякою перешкодою при написанні переносних програми. Python ж таким недоліком не володіє.

Наступною важливою характеристикою є масштабованість мови. Це означає, що є можливість вдосконалення мови зацікавленими програмістами. Інтерпретатор написаний на C і вихідний код доступний для будь-яких маніпуляцій. У разі необхідності, можна вставити його в свою програму і

використовувати як вбудовану оболонку. Або ж, написавши на С свої доповнення до Python і скомпілювавши програму, отримати "розширений" інтерпретатор з новими можливостями.

Наступна перевага – наявність великого числа програми модулів, що забезпечують різні додаткові можливості. Такі модулі пишуться на С і на самому Python і можуть бути розроблені усіма досить кваліфікованими програмістами. Це дозволяє не витрачати час на розробку вже давно готових модулів.

В якості СУБД вирішено обрати MySQL. MySQL підтримує SQL (структурована мова запитів) і може застосовуватися в якості SQL-сервера. Це означає, що спілкуватися з сервером можна на мові SQL: клієнт посилає серверу запит, той його обробляє і віддає клієнту тільки ті дані, які були отримані в результаті цього запиту.

Основними перевагами даної СУБД є:

1. Швидкодія. Завдяки внутрішньому механізму багатопоточності, швидкодія MySQL є дуже високою.
2. Безпека. Досить високий рівень безпеки забезпечується завдяки базі даних mysql, що створюється при установці пакету і містить п'ять таблиць. За допомогою цих таблиць можна описати, який користувач з якого домену з якою таблицею може працювати і які команди він може виконувати.
3. Ліцензія. Раніше ліцензування MySQL було трохи заплутаним; Зараз ця програма для некомерційних цілей поширюється безкоштовно.
4. Відкритість коду. Завдяки цьому є можливість самостійно додавати в пакет потрібні функції, розширюючи його функціональність так, як вам потрібно.
5. Надійність. Пакети MySQL є досить стабільні і їх важко вивести з ладу.
6. Універсальність. На сьогодні існують версії програми для більшості поширених комп'ютерних платформ та операційних систем.

Підсумовуючи приведену вище інформацію, варто зазначити, що реалізація інтелектуальної системи мерчандайзингу з використанням

можливостей мови програмування Python та СУБД MySQL поєднує в собі гнучкість, швидкість, використання мінімально кількості пам'яті, можливість легкої модифікації та крос платформність рішення.

3.2 Обґрунтування вибору архітектури класифікатора

Вибір архітектури класифікатора був найдовшим і найскладнішим етапом в процесі розробки класифікатора. Процес вибору архітектури потребує глибоких аналітичних знань і детального розуміння, як працює кожна з опрацьованих структур згорткових нейронних мереж. Рішення по вибору архітектури було прийнято після опрацювання більше 30 різноманітних варіацій структур і параметрів.

3.2.1 Архітектура глибокого мультикласового класифікатора на базі класичних згорткових мереж

Спочатку було прийняте рішення використовувати мультикласовий класифікатор. Беручи до уваги досягнення останніх років на конкурсах по машинній класифікації було прийняте рішення модифікувати уже існуючі рішення, які перемагали в престижному змаганні ILSVRC такі як AlexNet 2012 [19], GoogLeNet 2014 [20], ResNet (2015) [27]. Це щорічне змагання являє собою одне із найважливіших змагань по розпізнаванню зображень, його переможці створюють шедеври в цій сфері. Задачею є класифікація зображень в 1000 раніше заданих класів. Три вибрані рішення є гнучкими і дозволяють ефективно розпізнавати товари.

Після аналізу і проведення ряду експериментів було прийняте рішення модифікувати мережу AlexNet. Дана мережа є найменш глибокою і найменш вибагливою до ресурсів, це призводить до менш глибокої екстракції ознак, проте до набагато швидшого процесу навчання. Це дуже важливо, враховуючи специфічну задачу класифікації. Мережа AlexNet складається всього з 5 згорткових і 3 повнозв'язних шарів.

В запропонованій модифікації на вхід подавалось зображення розміром 512x512, що в два рази більше аналогічного показника в інших мережах. Було додано додатковий згортковий шар на початку мережі. Структура запропонованої мережі зображена на рисунку 3.1.

Також зменшено кількість параметрів до 33 мільйонів в порівнянні з 60 в AlexNet. Різні глобальні параметри було протестовано для визначення оптимальних. Для оптимізації використовувались три різні підходи: стохастичний градієнтний спуск, градієнт Нестерова і адаптивний градієнтний алгоритм. В AlexNet використовували батч в 256 зображень і швидкість навчання рівну 0.02, для запропонованої структури було вибрано батч в 128 зображень і швидкість навчання рівну 0.01. Модель навчалась протягом 150 епох.

Для навчання моделі використовувався python 3.6 з бекендом tensorflow 1.7.0. Навчання проводилось на базі GPU NVIDIA GeForce GTX 1080Ti, однієї з найпотужніших відеокарт доступних на ринку. Дана відеокарта побудована з використанням новітньої архітектури Pascal з 11 Gb GDDR5X пам'яті, пропускною здатністю в 11 Gbps і частотою 1582 MHz, проте навіть з такими даними вона навчала запропоновану мережу для класифікації товарів по 30 класах більше 12 годин.

Для тестування мережі було підготовано спеціальний пробний набір даних з різноманітними товарами з сфери ритейлу з інтернету. Результати роботи запропонованої мережі, а також порівняльна характеристика з трьома готовими мережами на пробному наборі наведені в таблиці 3.1.

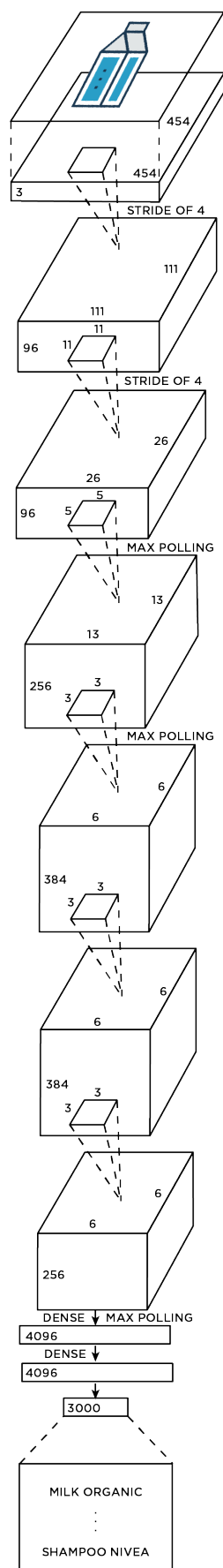


Рисунок 3.1 Структура запропонованої мережі

Таблиця 3.1 – Порівняльна характеристика роботи різних мереж

Модель	Тренувальний сет		Валідаційний сет		Тестовий сет	
	Loss	Acc.	Loss	Acc.	Loss	Acc.
AlexNet SGD	0.41	89.78	0.55	81.99	0.44	84.77
AlexNet NAG	0.33	89.90	0.57	82.04	0.43	84.12
AlexNet AdaGrad	0.52	89.37	0.61	80.20	0.46	84.89
GoogLeNet SGD	0.41	90.70	0.80	80.64	0.59	86.33
GoogLeNet NAG	0.36	90.73	0.79	80.66	0.59	86.12
GoogLeNet AdaGrad	0.48	87.50	0.75	81.24	0.47	86.64
ResNet SGD	0.64	81.86	0.34	85.42	0.28	87.77
ResNet NAG	0.43	83.88	0.27	85.08	0.26	87.99
ResNet AdaGrad	0.51	83.21	0.33	83.78	0.32	86.51
Proposed SGD	0.45	88.59	0.45	80.71	0.26	86.69
Proposed NAG	0.44	88.53	0.43	81.12	0.26	86.58
Proposed AdaGrad	0.44	89.76	0.45	80.65	0.25	87.48

Попри те, що було досягнуто точності класифікації в 87,48%, що являється найкращим результатом в порівнянні з іншими роботами в подібному полі [28 - 30], дана структура не підходить для вирішення поставленої задачі. В середньому в одному невеликому магазині знаходиться 6000 різних товарів, що робить використання запропонованої мережі майже неможливим в реальних умовах.

Враховуючи цей фактор від такої архітектури прийшлося відмовитись. Пошуки оптимальної архітектури продовжився з ухилом на оптимізацію швидкості навчання.

3.2.2 Модифікована мережа з модулем Inception

Після повторного аналізу трьох раніше описаних глибоких мереж було виявлено, що з мережі GoogLeNet можна взяти модуль під назвою Inception, який після відповідних модифікацій дозволив скоротити час навчання в 2 рази і розмір самої мережі в 10. В цьому підході розмір вхідного зображення склав 256x256 пікселів, як і в оригінальній GoogLeNet. Стандартний модуль Inception (Рисунок 3.2) було перероблено і запропоновано дві його модифікації для використання в запропонованій мережі (Рисунок 3.7-3.8).

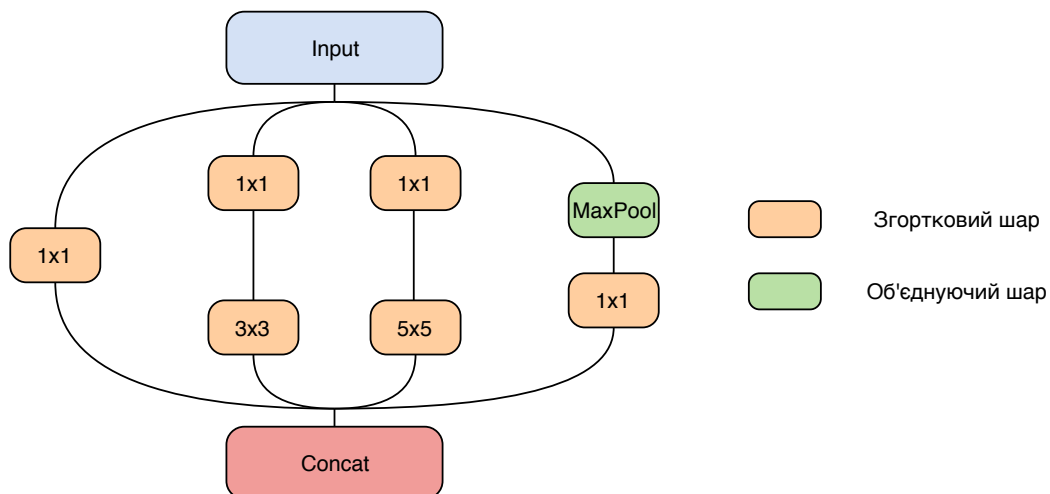


Рисунок 3.2 – Стандартний модуль Inception

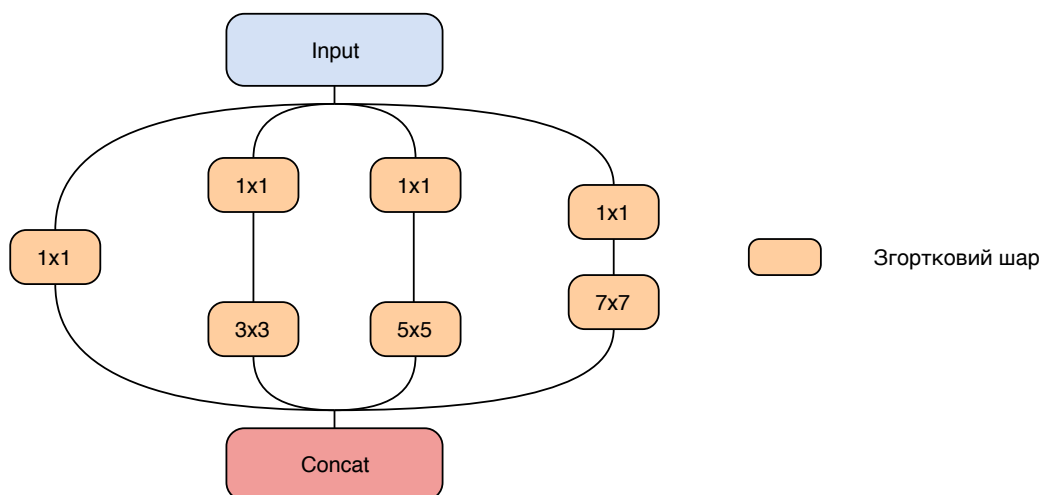


Рисунок 3.3 – Перша модифікація модуля Inception

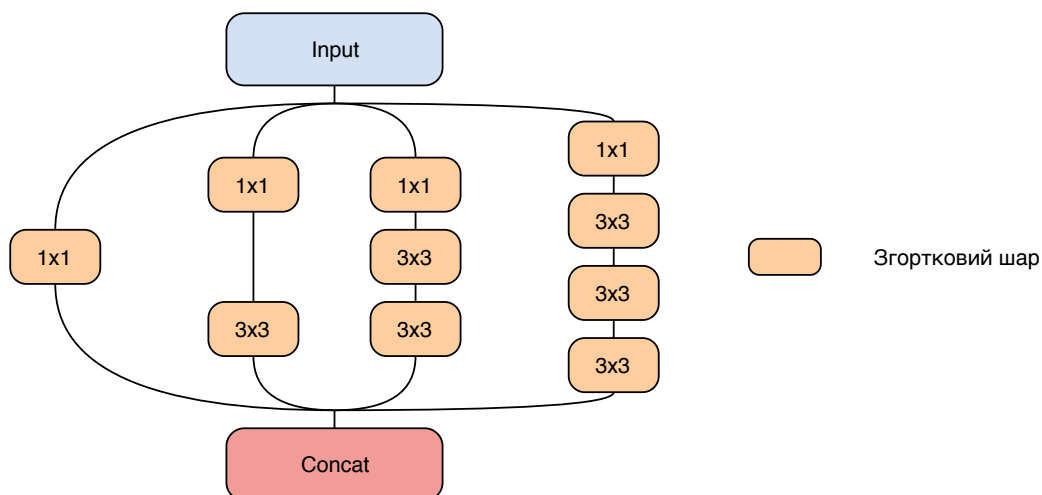


Рисунок 3.4 – Друга модифікація модуля Inception

У першій модифікації Inception згортковий шар 1×1 замінено шаром 7×7 , а об'єднуючий шар 3×3 на згортковий шар 1×1 . В другій модифікації згортковий шар 5×5 замінено на два шари 3×3 , а шар 7×7 на три шари 3×3 . Дві модифіковані версії модулів Inception були застосовані в двох модифікованих мережах.

Від мережі GoogLeNet (Рисунок 3.5) запропоновані мережі відрізняються в першу чергу використанням тільки одного модуля Inception, що дозволяє досягнути значно менших вимог до ресурсів необхідних для ефективного навчання.

Перша модифікована мережа (рисунок 3.6) повторює структуру GoogLeNet і відрізняється тільки модулем Inception. Детально шари мережі описані в Таблиці 3.2.

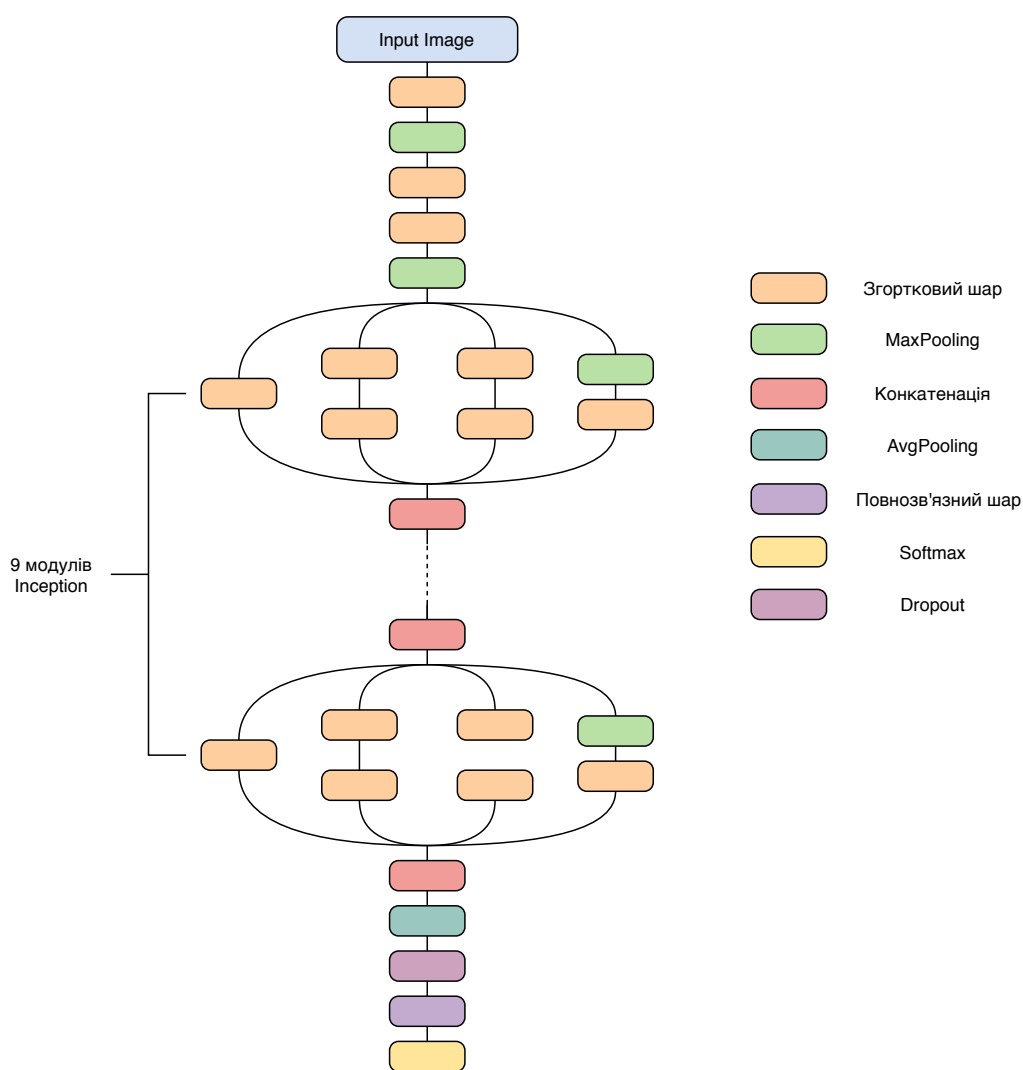


Рисунок 3.5 – Структура мережі GoogLeNet

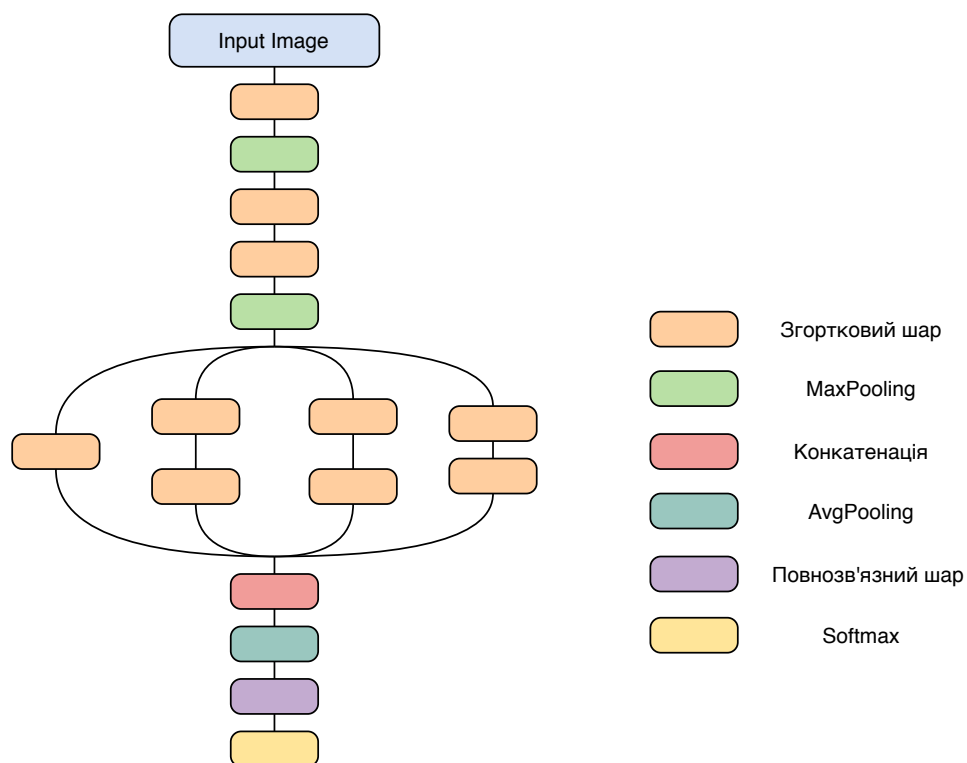


Рисунок 3.6 – Структура першої запропонованої мережі

Таблиця 3.2 – Опис шарів першої запропонованої мережі

Тип шару	Розмір ядра / крок	Вихідний розмір
Вхідне зображення	-	224x224x3
Згортковий	7x7/2	109x109x96
Об'єднуючий (макс)	3x3/2	54x54x96
Згортковий	1x1/1	54x54x96
Згортковий	5x5/2	25x25x96
Об'єднуючий (макс)	3x3/2	12x12x96
Модуль Inception ver. 1	Рисунок 3.3	12x12x256
Об'єднуючий (сер)	-	1x1x256

Друга модифікована мережа (Рисунок 3.7) відрізняється від GoogLeNet більш радикально. І в кінцевому результаті вона дає кращі результати точності чим перша, проте потребує більше часу для навчання. Детально шари мережі описані в Таблиці 3.3.

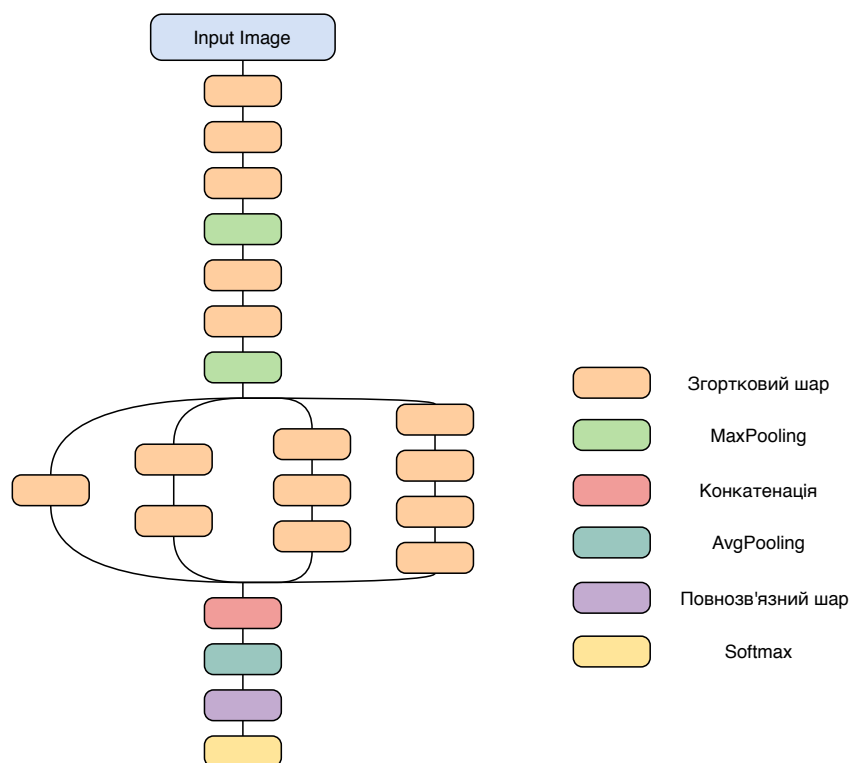


Рисунок 3.7 – Структура другої запропонованої мережі

Таблиця 3.3 – Опис шарів другої запропонованої мережі

Тип шару	Розмір ядра / крок	Вихідний розмір
Вхідне зображення	-	224x224x3
Згортковий	3x3/2	111x111x32
Згортковий	3x3/1	109x109x32
Згортковий	3x3/1	109x109x64
Об'єднуючий (макс)	3x3/2	54x54x64
Згортковий	1x1/1	54x54x64
Згортковий	5x5/2	25x25x96
Об'єднуючий (макс)	3x3/2	12x12x96
Модуль Inception ver. 2	Рисунок 3.4	12x12x256
Об'єднуючий (сер)	-	1x1x256

При навчанні мереж було використано адаптивний градієнтний алгоритм для оптимізації і наступні гіперпараметри: батч розміром 64 зображення, швидкість навчання 0.01. Навчання проводилось в 100 епох. Результати навчання запропонованих мереж і їх порівняння приведені в Таблиці 3.4. Дані результати приведені з ухилом в сторону швидкодії мережі, а не її точності, так як перед запропонованою архітектурою ставилась саме така задача. Мережа ResNet не приймала участь в порівнянні, так як час її навчання був дуже великим і порівняння з врахуванням поставленої задачі не було доцільним.

Таблиця 3.4 – Результати роботи запропонованих мереж

Модель	Точність	Швидкість (зображень/мс)	Кількість параметрів (млн)
AlexNet	86.78	13.8	58.48
GoogLeNet	91.94	40.36	10.45
Proposed ver.1	92.07	18.21	0.87
Proposed ver.2	92.89	37.48	0.88

Результати роботи виявились набагато кращими, чим очікувалось. Друга версія мережі досягнула точності в фантастичні 92.89 процента, при цьому маючи дуже високу швидкодію. Попри суттєве покращення результатів в порівнянні з модифікацією AlexNet, нова структура також навчалась дуже довго, що унеможливлювало її використання для вирішення поставленої задачі.

3.2.3 Запропонована архітектура з бінарним класифікатором

Незважаючи на дуже високі показники точності класифікації у раніше розглянутих архітектурних рішень критична проблема залишається невирішеною. Обидва підходи дуже довго навчаються. Додатково взявши до

уваги те, що в середньому в одному невеликому магазині ритейлера представлено від 4 до 6 тисяч товарних позицій використання мультикласового класифікатора для вирішення поставленої задачі стає майже неможливим.

Враховуючи всі обмеження було прийняте рішення відмовитись від складних мультикласових рішень і спробувати вирішити задачу в вигляді бінарної класифікації. Так для кожного унікального товару створюється свій простий класифікатор, який на виході визначає ймовірність приналежності товару до класу “об’єкт”. Структура класифікатора обрана максимально проста, але з врахуванням усіх особливостей і налаштована таким чином, щоб ефективно класифікувати всі представлені товари в ритейлі. Ваги кожного навченого класифікатора зберігаються в базі даних і загружаються кожний раз на вимогу. Це дозволяє швидко і ефективно класифікувати великі кількості товарів, не витрачаючи захмарних ресурсів.

З урахуванням проаналізованої інформації, було запропоновано використання класифікатора, який детально описаний в розділі 2.

3.3 Аналіз архітектури бази даних

База даних для інтелектуальної системи мерчандайзингу складається з таких основних компонент (Рисунок 3.8):

- База даних для навчання нейронної мережі. Зберігає набори даних по товарним групам та самих товарів для навчання та перевірки нейронної мережі для класифікації товарів (Рисунок 3.9).
- База даних зображень для обробки та аналізу. Оскільки система працює в реальному часі та на всіх ТТ ритейлера, то доцільно зберігати зображення для аналізу, щоб уникнути проблем вразі аварійного припинення роботи, або ж для розподілу навантаження на систему (Рисунок 3.10).

- База даних довідників та метаданих. Дана база даних необхідна для коректного відображення даних в інтерфейсній частині для взаємодії з клієнтів, також в ній зберігаються дані про розташування камер, полиці та їх плани, що є необхідною інформацією для роботи системи (Рисунок 3.11).
- База даних результатів. В даній базі даних відбувається накопичення історичних результатів по аналізу полиць, збір статистики для подальшого використання та побудови звітності (Рисунок 3.12).

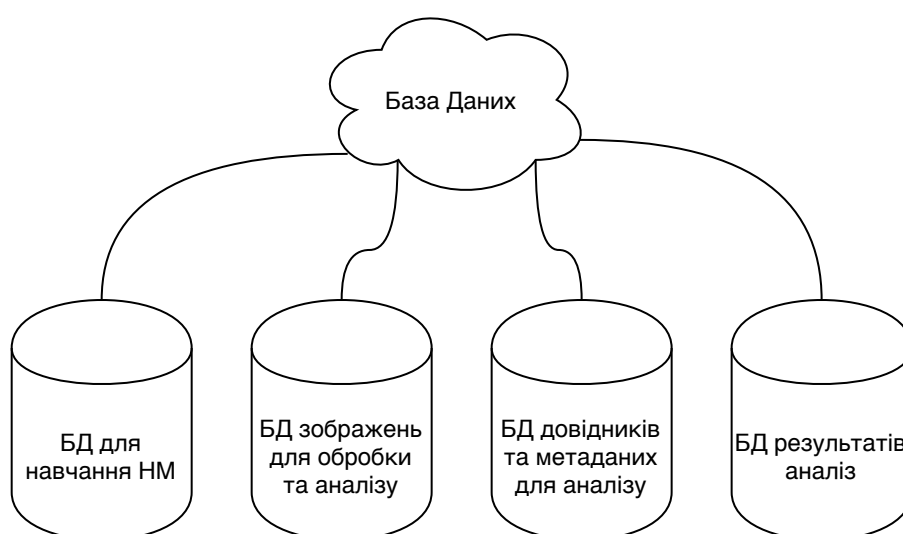


Рисунок 3.8 – Узагальнена діаграма структури БД

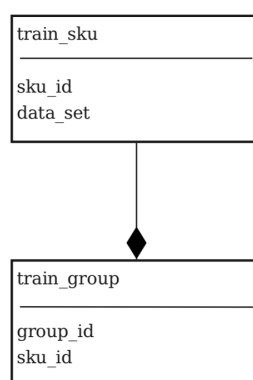


Рисунок 3.9 – Діаграма БД для навчання НМ

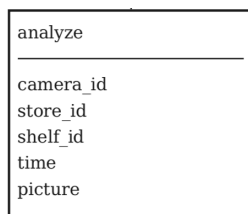


Рисунок 3.10 Діаграма БД зображень для аналізу

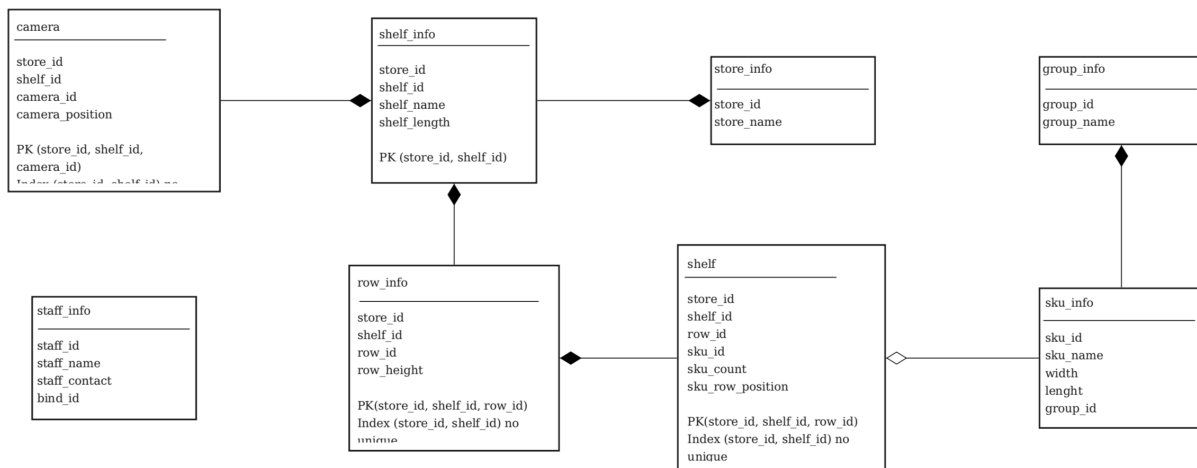
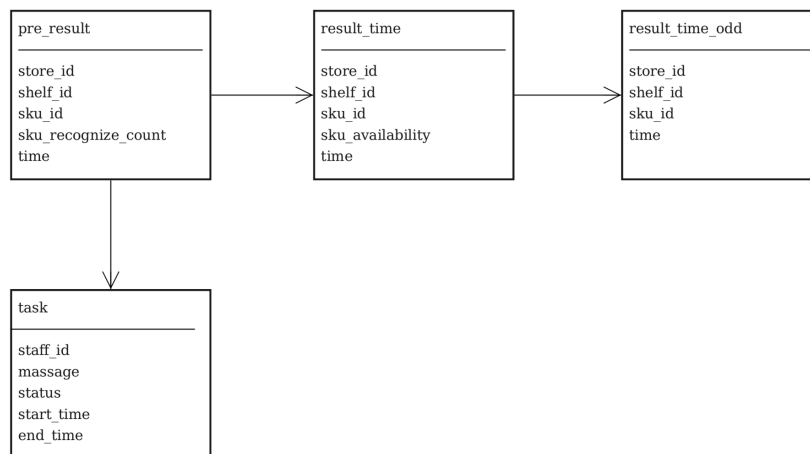


Рисунок 3.11 – Діграма БД словників та метаданих



3.12 Діаграма БД результатів

3.4 Аналіз архітектури програмного продукту

Програмний продукт інтелектуальної системи мерчандайзингу складається з наступних модулів (Рисунок 3.12):

1. модуль управління камерами в ТТ, зчитування зображень в реальному часі;
2. модуль роботи з БД;
3. модуль сегментації зображень;
4. модуль класифікації товарів;
5. модуль аналізу результатів;
6. модуль взаємодії з персоналом.



Рисунок 3.12 – Схема модулів інтелектуальної системи мерчандайзингу

Модуль управління камерами в ТТ, зчитування зображень в реальному часі відповідає за управління камерами, зчитування з них зображень, співставлення зображень з камер з фізичними полицями та відправка цих даних на центральний

сервер для подальшого аналізу. Даний модуль представляє собою набір камер, сервер, до якого підключені камери та бібліотеку управління камерами.

Модуль роботи з базами даних є набором процедур та функцій, за допомогою яких виконується вивантаження та завантаження даних в БД.

Метою модуля сегментації зображень є сегментація зображення та мапінг товарів відносно планограми. Даний модуль складається з наступних бібліотек:

1. бібліотека мапінгу товарів;
2. бібліотека методу Mean shift;
3. бібліотека методу порогового значення;
4. бібліотека методу водорозділу.

Модуль класифікатора товарів складається з нейронної мережі та хмари класифікаторів. Даний модуль використовується для класифікації попередньо сегментованих товарів для подальшого співставлення класифікованих товарів відповідно товарам планограми для аналізу якості викладки товарів.

Модуль аналізу результатів представляє собою набір функцій для аналізу історичних даних про наповнення полиць, якості викладки відповідно планограм. Результатами роботи даного модулю є статистичні дані аналізу зображень полиць та якості роботи персоналу.

Модуль взаємодії з персоналом являє собою набір веб сторінок для взаємодії з інтелектуальною системою мерчандайзингу. Дозволяє отримувати максимальний обсяг інформації, звітів необхідних для роботи з системою.

Висновки за розділом 3

В даному розділі було аргументовано вибір програмної реалізації інтелектуальної системи мерчандайзингу та мови програмування для розробки системи. Інтелектуальна система мерчандайзингу реалізована як набір окремих

модулів за допомогою мови програмування Python. Для збереження та систематизації даних було обрано систему управління базами даних MySQL.

Описано розглянуті підходи до побудови і реалізації класифікатора, аргументовано вибір кінцевої архітектури для класифікатора.

Наведено архітектуру реалізації бази даних для інтелектуальної системи мерчандайзингу. Обґрунтовано та описано вибір даної архітектури бази даних.

Розглянуто програмну архітектуру реалізації інтелектуальної системи мерчандайзингу та наведено опис основних модулів, їх призначення та функціональні можливості.

РОЗДІЛ 4. АНАЛІЗ РЕЗУЛЬТАТІВ

4.1 Аналіз роботи алгоритму класифікації

Складність алгоритму – це основний показник ефективності роботи та оптимальності реалізації алгоритму. Складність алгоритму дає можливість дослідити алгоритм, знайти слабкі та сильні сторони та дослідити можливості оптимізації. Алгоритм класифікації зображення базується на згортковій нейронній мережі. В випадку з нейронними мережами варто розглядати дві складності – обчислювальну і часову. Часова складність напряму залежить від заліза, тому при оцінці якості алгоритму за цим критерієм потрібно проводити порівняння з іншими алгоритмами на однакових машинах. Запропонований класифікатор складається з шарів згортки і шарів класифікації. Складність шарів класифікації відповідає складності стандартного перцептрона. Тому складність алгоритму можна розглядати як складність згорткової частини і частини класифікатора.

Для оцінки складності згорткової частини використаємо наступні позначення:

- d – глибина мережі (кількість згорткових шарів)
- n_l – кількість фільтрів в шарі l
- n_{l-1} – кількість вхідних ознак в шар l
- s_l – розмірність ядра

Загальна часова складність всіх згорткових обраховується по формулі:

$$O\left(\sum_{l=1}^d n_{l-1} \cdot s_l^2 \cdot n_l\right) \quad (4.2)$$

Застосувавши дану формулу до запропонованої мережі можна легко обрахувати складність згорткових шарів:

$$complexity = 512 \cdot 3^2 \cdot 32 + 223 \cdot 3^2 \cdot 32 + 124 \cdot 3^2 \cdot 64 \quad (4.2)$$

В таблиці 4.1 наведено порівняльні результати складності запропонованого алгоритму і алгоритмів загальновідомих CNN для класифікації. Для наочного порівняння складність запропонованого алгоритму була взята за одиницю, а складності інших мереж показують в скільки разів вони відрізняються від запропонованої.

Таблиця 4.1 – Порівняння складностей CNN

Модель	Складність	Секунд/Батч
AlexNet	9.7	0.5
VGG-16	28.3	9.6
GoogLeNet	4.1	3.2
Запропонована	1	0.21

Беручи до уваги результати, наведені в таблиці 4.1, можна зробити висновки, що запропонований алгоритм є швидким і ефективним, перевершує показники існуючих мереж і підходить для вирішення поставленої задачі класифікації.

4.2 Аналіз якості роботи системи класифікації

Аналіз якості роботи системи класифікації поділений на два етапи і проводиться на тестовому зразку “Йогурт Органік з полуницею”:

- Оцінка якості навчання запропонованого класифікатора;
- Оцінка якості класифікації на зображеннях, які не ввійшли в вибірки.

4.2.1 Аналіз якості навчання

Аналіз якості навчання полягає в оцінці якості навчання, а саме точності на кожній вибірці, а також оцінці якості процесу навчання. Якість процесу навчання можна оцінити по ціні навчання на певній кількості епох. Ціна навчання повністю залежить в першу чергу від обраного методу оптимізації.

На рисунку 4.1 зображено порівняльну характеристику якості процесу навчання з використанням різних методів оптимізації на перших трьох епохах для запропонованої структури згорткової нейронної мережі на загальнодоступній базі зображень CIFAR10.

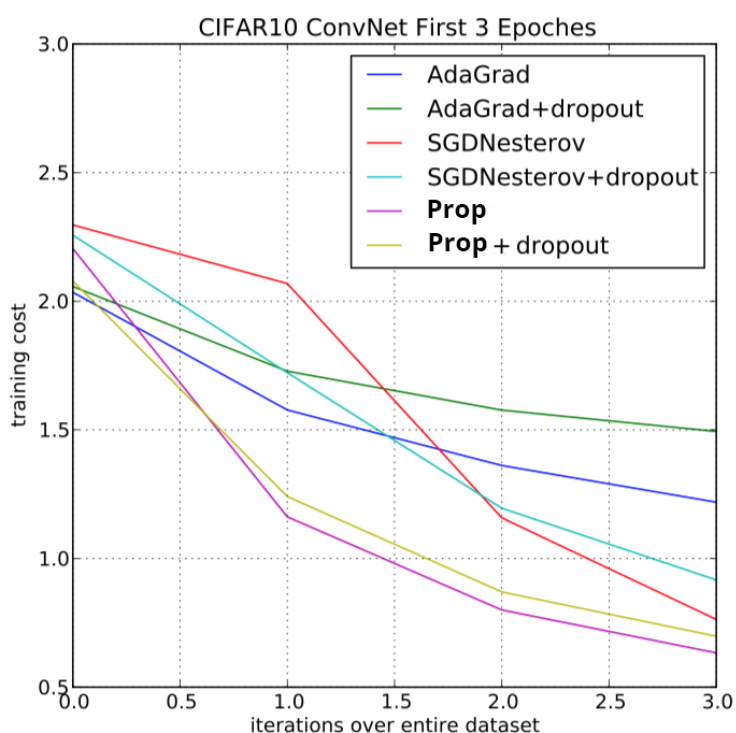


Рисунок 4.1 – Якість навчання на перших 3 епохах

Як можна помітити, запропонований метод (на рисунку 4.1 відображений як Prop – proposed – запропонований) дає кращі результати чим інші загальновідомі підходи до оптимізації, а саме AdaGrad (метод адаптивного стохастичного градієнтного спуску) і SGD Nesterov (метод стохастичного

градієнтного спуску з використанням моменту Нестерова). Незалежно від того, чи було використано техніку dropout чи ні.

Аналогічні дані для 45 епох навчання приведені на рисунку 4.2

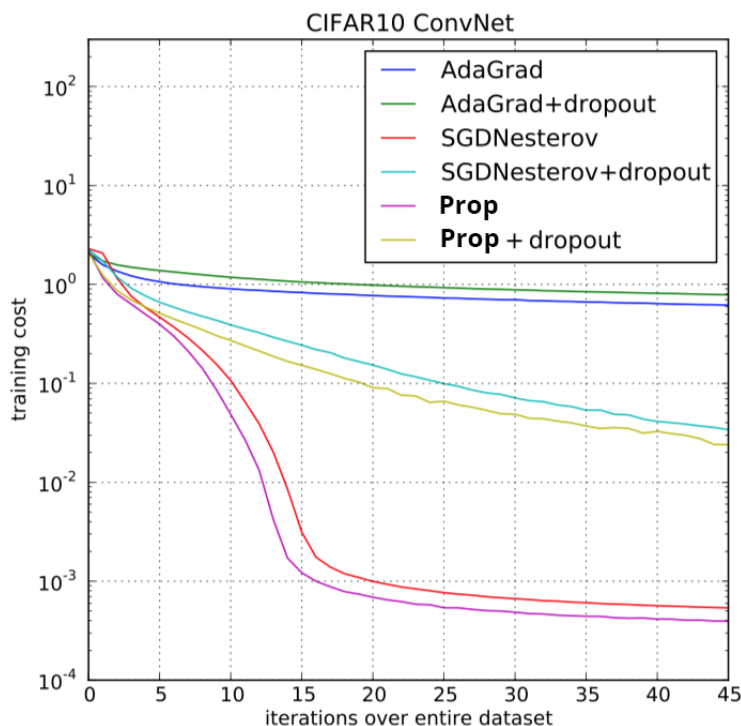


Рисунок 4.2 – Якість навчання на 45 епохах

На повному наборі епох запропонований метод також показує кращі результати, чим його конкуренти.

З отриманих результатів видно, що процес навчання є ефективним.

На рисунку 4.3 зображено графік зміни ціни, а також точності на тренувальній і валідаційній вибірках. Для кращого графічного представлення графік побудований на 25 епох:

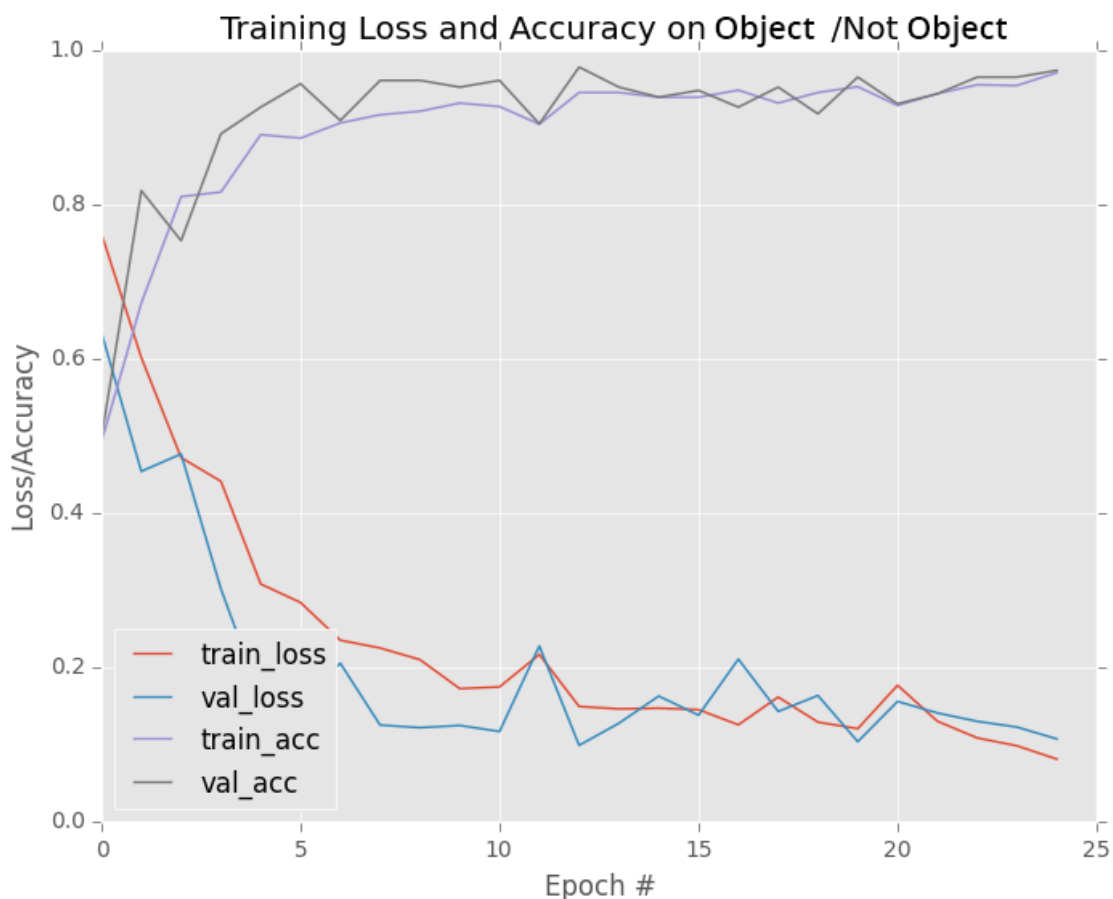


Рисунок 4.3 – Графік зміни ціни і точності

Як видно з графіка, система достатньо швидко навчається. Якість навчання мережі є задовільною.

4.2.2 Аналіз якості класифікації

Якість класифікації варто оцінювати за показником середньої точності на вибірках зображень, які не приймали участь в класифікації. Точність розглядається в двох інтерпретаціях:

1. усереднені ймовірнісні значення, які отримувались на виході мережі, як при правильній, так і не правильній класифікації;

2. відсоток правильно класифікованих об'єктів, відносно всіх об'єктів, що приймали участь в класифікації.

На вхід системи подавались тестові зображення з різних товарних груп середньостатистичного ритейлера. В системі ініціалізували ваги для кожного товару і оцінювали якість класифікації по вихідних даних. На виході система надає ймовірність приналежності товару то до класу “об'єкт”, або класу “не об'єкт”.

Для того, щоб оцінку якості можна було вважати прийнятною обрано 100 найпоширеніших товарів з 20 товарних груп. Проаналізовано результати роботи системи з запропонованою архітектурою, а також системи з 5 подібними архітектурами. Результатати порівняння наведені в таблиці 4.2. Відображено структуру мережі, середнє значення ймовірностей, кількість правильно і неправильно класифікованих товарів, середнє точність класифікації.

Таблиця 4.2 – Порівняльна характеристика різних структур з запропонованою

Model	Avg. Prob.	Pos.	Neg.	Avg. Acc.
32c+p – 32c+p – 128fc – 1fc	66.9	88	12	88%
32c+p – 32c+p – 64c+p – 128fc+dropout – 1fc	75.1	96	4	96%
32c+p – 32c+p – 32c+p – 64c+p – 128fc – 1fc	69.4	86	14	86%
32c+p – 32c+p – 32c+p – 64c+p – 64c+p – 128fc – 1fc	68.2	89	11	89%
32c+p – 32c+p – 32c+p – 64c+p – 128fc+dropout – 1fc	73.89	92	8	92%
32c+p – 32c+p – 32c+p – 64c+p – 64c+p – 128fc+dropout – 1fc	75.47	93	7	93%

Для більш наочного сприйняття результатів, розгорнуті результати роботи запропонованої системи приведені в таблиці 4.3. Представлені дані агреговані по групах.

Таблиця 4.3 – Результати роботи системи класифікації на тестовій вибірці

Group	Avg. Prob., %	Positive Count	Negative Count	Avg. Acc., %
Молоко	69.8	5	0	100
Йогурти	75.4	5	0	100
Пиво	73.2	5	0	100
Сидр	73.1	5	0	100
Кава	82.7	5	0	100
Чай	63.4	3	2	60
Крупи	65.9	5	0	100
Соки	88.3	5	0	100
Цукор	71.2	5	0	100
Сіль	72.3	5	0	100
Мука	69.9	4	1	80
Вода сол.	83.4	5	0	100
Горілка	84.6	5	0	100
Консерви	81.2	5	0	100
Шоколад	68.7	5	0	100
Конд. Фас.	60.3	4	1	80
Сухі снід.	91.4	5	0	100
Макарони	73.6	5	0	100
Вода мін.	88.12	5	0	100
Цукерки	65.3	5	0	100
Σ	75.1	96	4	96

Як видно з таблиці, попри не дуже високу ймовірнісну точність, основна точність класифікації дуже висока. Саме ця точність є основним критерієм оцінки якості роботи системи, так як основною задачею є точна класифікація, а не досягнення високих показників ймовірності приналежності класу.

Для оцінки якості системи було заповнено тестову частину БД, навчено класифікатор на тестовій групі товарів, а також внесено планограму магазину (розташування полиць). Планограма магазину зображена на рисунку 4.5.

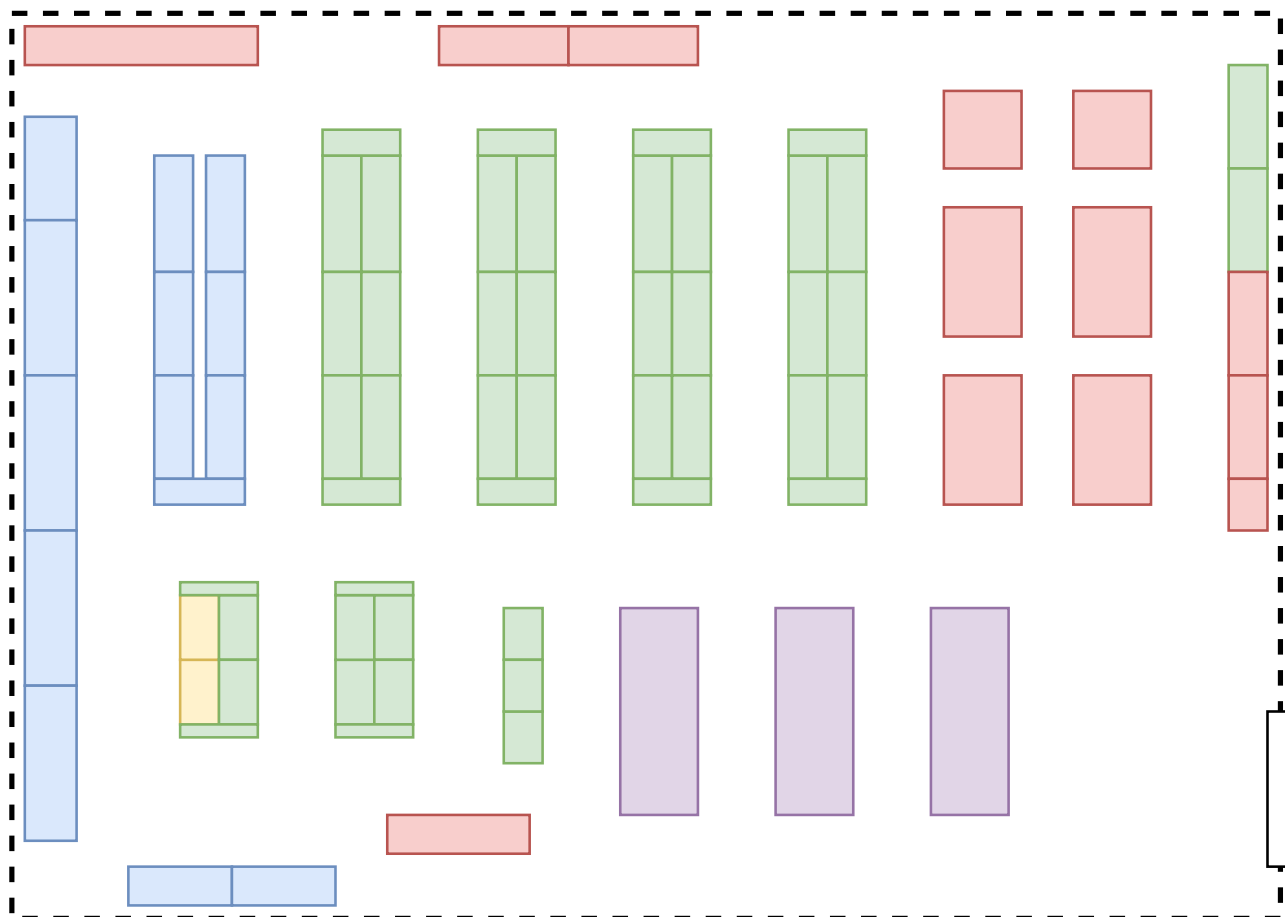


Рисунок 4.5 – Планограма магазину

На планограмі (Рисунок 4.5) зображено реальне розташування полиць в реальному магазині. Кольори полиць відповідають за наступне:

1. червоний – розсипні товари, та інші товари з якими система не працює, тому ці полиці в оцінці якості не приймають участь;
2. синій – холодильні камери, система здатна частково працювати з товарами, які знаходяться в холодильних камерах;
3. зелений – всі інші товари, з якими система працює в штатному режимі;
4. білий – білим квадратом справа позначено вхід в магазин;
5. фіолетовий – позначення кас для розрахунку за товари;
6. жовтий – позначає яйця, система працює в штатному режимі.

На рисунку 4.6 зображено розташування камер в магазині для отримання даних щодо розташування товарів на полицях. В залежності від розмірів полиці використовується одна, дві, три або чотири камери.

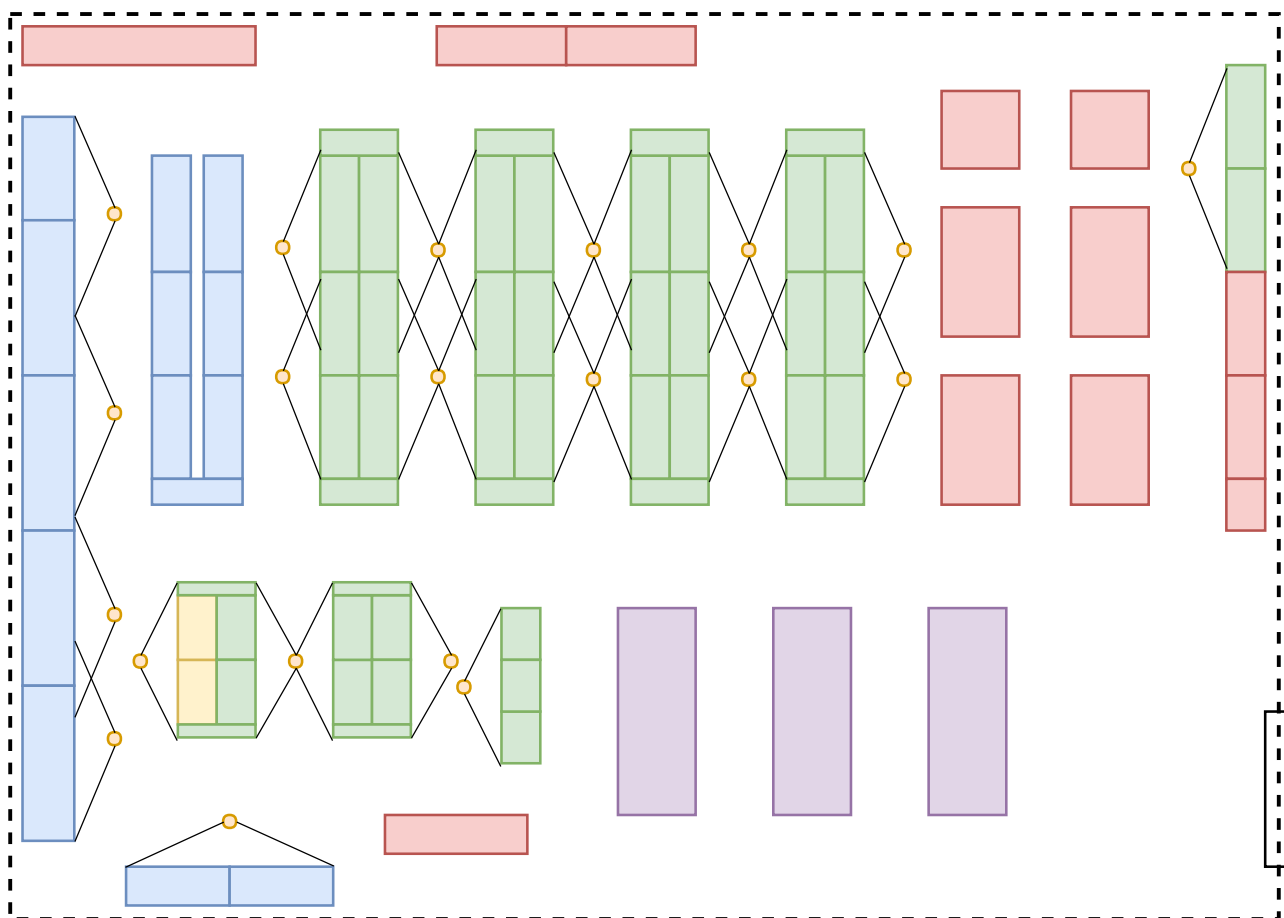


Рисунок 4.6 – Схема розташування камер

Камери розташовуються таким чином, щоб охопити весь простір полиці. Кількість камер залежить від фізичних розмірів полиці і специфіки товарів, розташованих на полиці.

Дані з камер передаються в систему. Далше система проводить автоматичне визначення знаходження товарів за допомогою системи сегментації

Стандартний вигляд полиці з товарами зображений на рисунку 4.7.



Рисунок 4.7 – Стандартний вигляд полиці

Всі наступні дії системи не відображаються для користувача. Користувач автоматично отримує дані, щодо товарів на полиці і індивідуально налаштовану аналітику. Користувач має можливість подивитись реальні дані з будь-якої камери в магазині і оцінити якість роботи системи.

Аналіз роботи системи класифікації і сегментації приведені раніше у відповідних розділах роботи.

На Рисунку 4.8 зображено вигляд системи для користувача. Зображено планограму, і аналітичні дані по магазину.

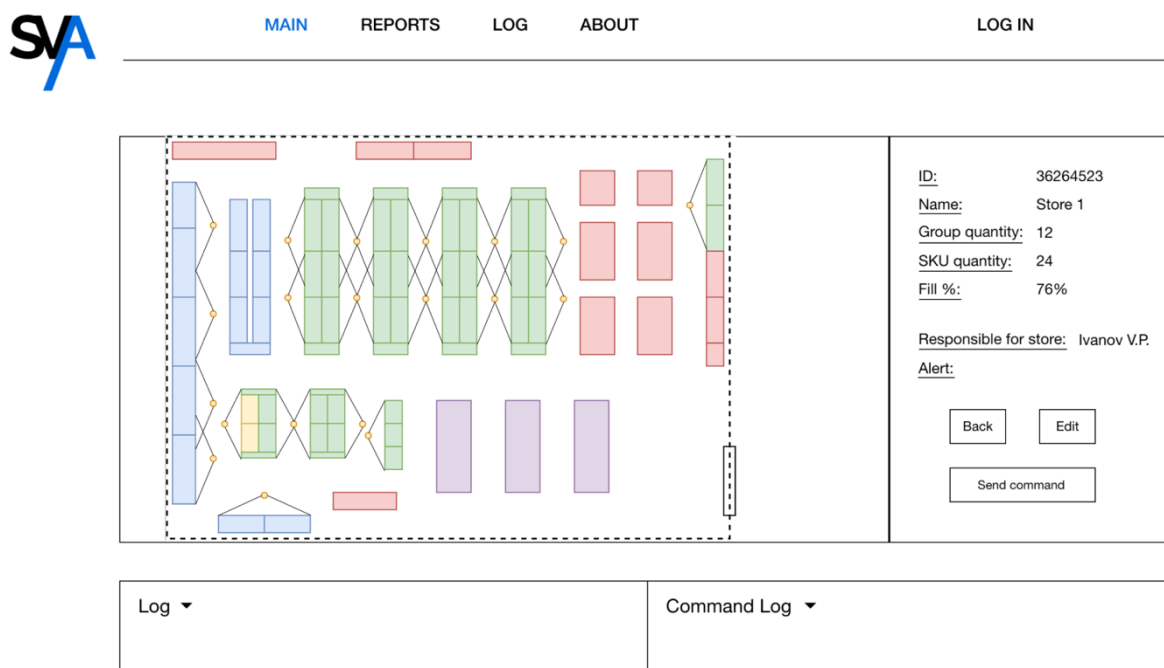


Рисунок 4.8 – Система в розрізі магазину

В системі в розрізі магазину вказано відповідального за полицю, кількість товарних груп, кількість товарних позицій, назву і унікальний ідентифікатор. Також відображається відсоток заповненості магазину. Дані отримані системою порівнювались з реальними даними по магазину. Реальні дані по заповненості товаром магазину відрізнялись від даних отриманих системою на 3-5%, що являється дуже хорошим показником.

Робота системи в розрізі одного стелажу зображена на рисунку 4.9.




	<p>ID: 36264577</p> <p>Name: Shelf 1</p> <p>Group quantity: 3</p> <p>SKU quantity: 6</p> <p>Fill %: 71%</p> <p>Responsible for shelf: Petrov I.O.</p> <p>Alert:</p> <p>Back Edit</p> <p>Send command</p>
Log ▼	Command Log ▼

Рисунок 4.9 – Система в розрізі стелажу

В розрізі стелажу користувач бачить товари і відсоток заповнення. Дані отримані системою порівнювались з реальними даними по стелажу в магазині. Реальні дані по заповненості товаром стелажу відрізнялись від даних отриманих системою на 2-3%, що являється дуже хорошим показником.

Робота системи в розрізі однієї полиці зображена на рисунку 4.10.




		<p>ID: 36264578</p> <p>Name: Group 1</p> <p>SKU quantity: 2</p> <p>Fill %: 71%</p> <p>Responsible for group: Fedorov P.I.</p> <p>Alert:</p> <p>Back Edit</p> <p>Send command</p>
Log ▼	Command Log ▼	

Рисунок 4.10 – Система в розрізі полиці

В розрізі полиці користувач бачить товари розташовані на полиці і відсоток заповнення. Дані отримані системою порівнювались з реальними даними по полиці в магазині. Реальні дані по заповненості товаром полиці відрізнялись від даних отриманих системою на 2-3%, що являється дуже хорошим показником.

Проаналізувавши дані, отримані системою при роботі на тестових групах товарів (було вибрано 10% усіх полиць магазину) і порівнявши їх з реальними даними в магазині було визначено, що точність роботи системи складає 95 – 97 відсотків, що являється дуже високим показником.

4.4 Керівництво користувача

Для роботи з системою розроблено спеціальний додаток “Мій кабінет” для користувача. В додатку користувач може проводити всі основні операції з

системою. Дії по адмініструванню системи дозволені тільки технічним працівникам, відповідальним за коректну роботу системи. Інтерфейс побудований таким чином, що користувач не може нашкодити роботі системи.

Початкове вікно додатку мій кабінет зображене на рисунку 4.11.

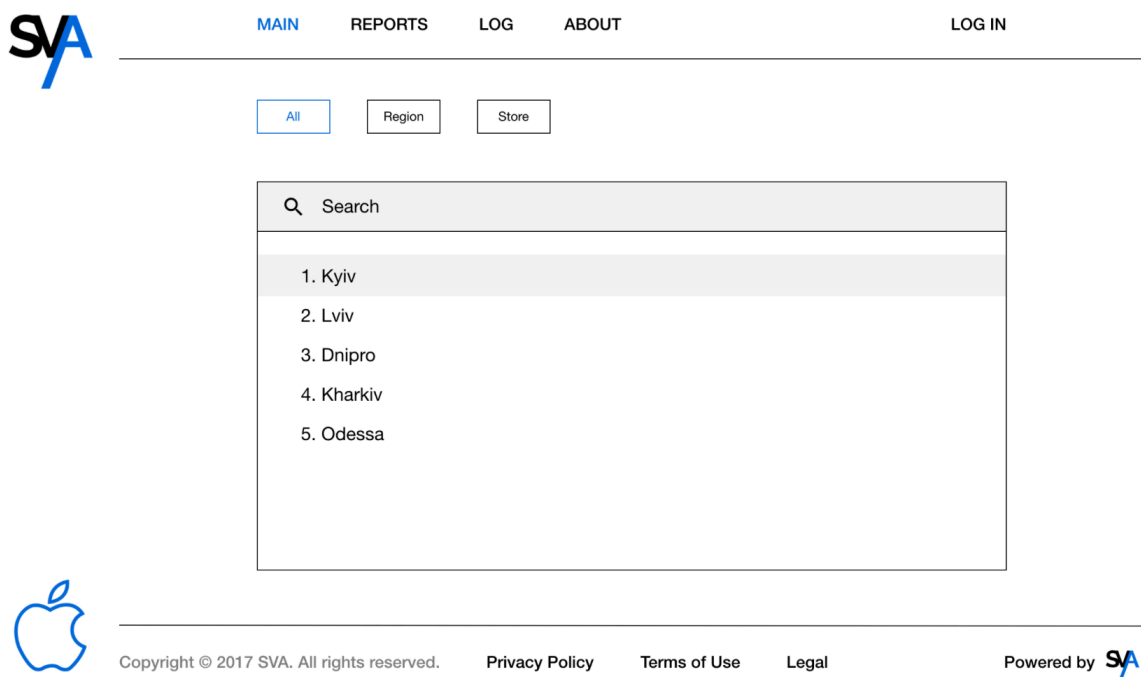


Рисунок 4.11 – Початковий екран власного кабінету

Для отримання доступу до конкретного магазину користувач повинен обрати місто, район і магазин. Вибір відбувається шляхом натискання на назву.

Після натиску на потрібну назву користувач отримує інформацію про обрану одиницю (Рисунок 4.12).



All

Region

Store

Q

Search

1. Kyiv

2. Lviv

3. Dnipro

4. Kharkiv

5. Odessa

ID:

873754048

Name:

Kyiv

Regions:

Shevchemkiivskiy district,
Solomenskiy district, Pecherskiy district.

Info:

Info info info info info info info info
info info info info info info info info
info info info info info info info.

Show regions



Рисунок 4.12 – Інформація про обраний об'єкт

Для вибору об'єкту необхідно натиснути на кнопку, виділену на рисунку 4.13.



All

Region

Store

Search

1. Kyiv

2. Lviv

3. Dnipro

4. Kharkiv

5. Odessa

ID:

873754048

Name:

Kyiv

Regions:

Shevchenkivskiy district,
Solomenskiy district, Pecherskiy district.

Info:

Info info info info info info info info
info info info info info info info info
info info info info info info info info
info info info info info info info.

Show regions



Рисунок 4.13 – Кнопка вибору об'єкта

Після вибору конкретного магазину користувачу відкривається планограма (Рисунок 4.14). На планограмі відображаються дані про магазин.

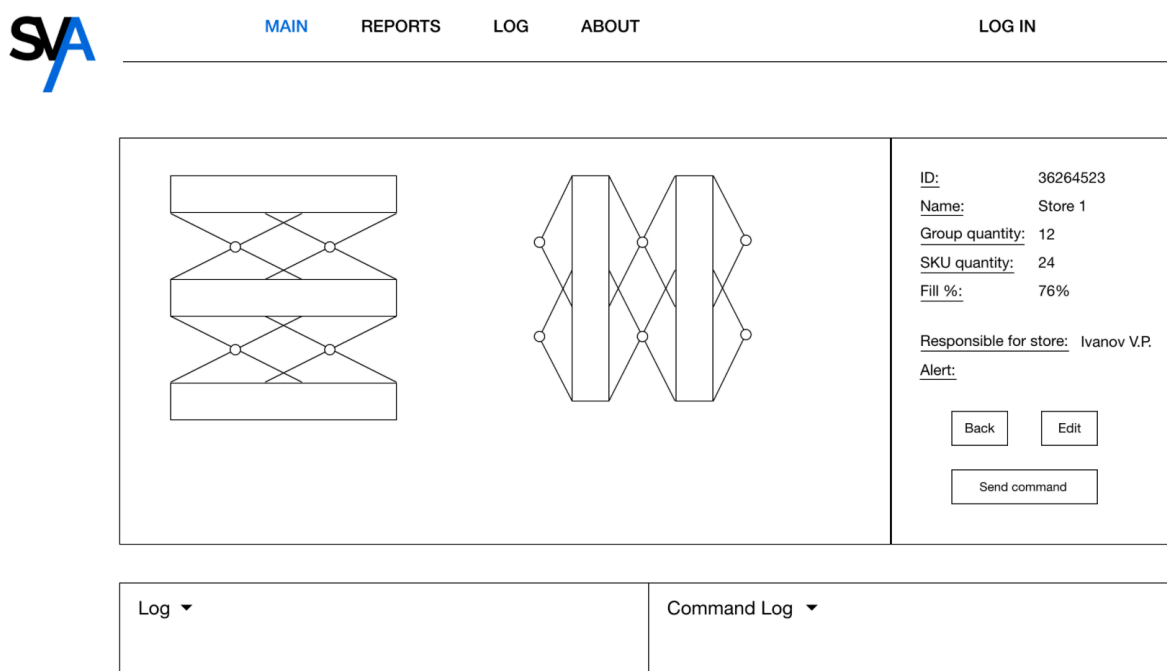


Рисунок 4.14 – Планограма магазину в кабінеті користувача

Для вибору стелажу необхідно натиснути на схематичне зображення потрібного стелажу.

В системі користувачу доступний наступний функціонал:

1. можливість редагувати планограму;
2. можливість в режимі реального часу отримати зображення з обраної камери;
3. відображення даних про обраний об'єкт;
4. відображення даних про відповідального за об'єкт;
5. відображення сповіщень про зміни в товарній викладці;
6. можливість відправити команду на поповнення полиці (Рисунок 4.15);
7. відображення логу роботи системи, а також логу виконання команд.

Send command

Responsible: Input Name ▼

Input command text

Back Send command

Command Log ▼

Рисунок 4.15 – Вікно відправки команд

Описаний вище функціонал доступний користувачу на всіх рівнях.

Для отримання детальних звітів по діяльності магазину та обслуговуючого персоналу наявний функціонал для генерації звітів.

Розділ кабінету для побудови звітів зображено на рисунку 4.16.

Інтерфейс кабінету користувача розроблено з врахуванням усіх потреб реальних користувачів. Функціонал кабінету повністю покриває всі можливості системи.



Category:

Keyword:

Start Date:

End Date:

Search



Рисунок 4.16 – Сторінка побудови звітів

Висновки за розділом 4

В даному розділі було проведено детальний аналіз роботи запропонованого алгоритму системи класифікації зображень. Досліджено його складність та порівняно її зі складністю подібних алгоритмів.

Наведено результати апробації роботи системи класифікації товарів на реальних даних з торгової точки. Оцінено якість класифікації на тестовій вибірці в 100 товарних одиниць з 20 товарних груп полиць товарів. Наведено результати тестової класифікації. Оціночна точність класифікації 97%.

Наведено результати апробації роботи інтелектуальної системи мерчандайзингу на реальній даних ТТ. Описано кожний етап роботи системи. Опис роботи інтелектуальної системи мерчандайзингу супроводжується візуальним зображенням результатів роботи кожного етапу роботи системи.

Проаналізовано якість роботи системи, яка складає 95 – 97 відсотків, що є дуже хорошим результатом.

Описано інструкцію користувача по роботі з системою. Розглянуто можливі варіанти взаємодії користувача з інтерфейсною частиною інтелектуальної системи мерчандайзингу.

РОЗДІЛ 5. СТАРТАП

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Узагальнено етапи розроблення стартап-проекту можна подати таким чином [31].

5.1 Опис ідеї проекту

Наведемо зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів у Таблиці 5.1.

Також проаналізуємо потенційні техніко-економічні переваги ідеї порівняно із пропозиціями конкурентів (Таблиця 5.2), оскільки перелік слабких, сильних та нейтральних характеристик та властивостей ідеї потенційного товару є підґрунтям для формування його конкурентоспроможності.

Таблиця 5.1 – Опис ідеї стартап проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система, що дозволяє моніторити наявність товарів на полицях в магазинах і перевіряє їх відповідність асортименту	Комерційний	Можливість контролювати повноту і точність асортименту магазину, що в свою чергу приводить до підвищення прибутку. Зменшення витрат на найм персоналу, мінімізація людського фактору.

Таблиця 5.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/ п	Техніко– економічні характеристи ки ідеї	(потенційні) товари/концеп ції конкурентів (конкурентів поки немає)	W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
1.	Система детекції і розпізнаванн я	Система працює на базі попередньо навченого класифікатора	Неможливіс ть класифікува ти об’єкт через перешкоди	Пропуск позицій класифікатор ом	Дуже висока точність класифікації незалежно від умов освітлення і розміщення
2.	Оновлення даних	Для роботи системи потрібно постійно актуальні дані	Зміни в асортименті можуть бути не вказані	Деякі незначні зміни не впливають на роботу	Можливість автоматично го підтягуванн я даних з бази рітейлера

5.2 Технологічний аудит ідеї проекту

Проведемо аудит технологій реалізацій основних ідей проекту. Для цього визначаємо технології та проведемо їх аналіз (Таблиця 5.3). Технології наведені в

Таблиці 5.3 – обрані для технічної реалізації проекту, а саме: методи сегментації і класифікації зображень і згорткові НМ та автоматичне оновлення даних з БД.

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Детекція і розпізнавання	Методи сегментації і класифікації зображень	Наявні, але варто покращити	Доступні авторам проекту.
2	Оновлення даних	Автоматично з бази, попередньо оновлено вручну	Наявна можливість оновлювати дані	Доступна

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначимо ринкові можливості, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту.

Для цього спочатку проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (Таблиця 5.4). Далі визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до товару для кожної групи (Таблиця 5.5).

Також проведемо аналіз ринкового середовища: факторів, що сприяють ринковому впровадженню проекту (Таблиця 5.6), та факторів, що йому перешкоджають (Таблиця 5.7). Та визначимо загальні риси конкуренції на ринку (Таблиця 5.8) та галузі (Таблиця 5.9).

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	1
2	Загальний обсяг продаж, грн/ум.од	1600000 ум.од
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	70%

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Підтримання встановленого асортименту магазину в якісній і кількісній складових.	Власники рітейл бізнесу	Відсутні	<ul style="list-style-type: none"> – Точність оцінки асортименту – Зручна звітність – Нативний інтерфейс

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Поява конкурентів	Конкуренти можуть зайняти велику частку ринку.	Розширення функціоналу, підвищення точності, підвищення лояльності.
2	Неможливість надавати якісні послуги	Клієнти можуть відмовитись від послуг в зв'язку з низькою якістю	Підвищення якості послуг, дії з повернення довіри клієнтів

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Розширення ринку рітейла	Поява нових клієнтів	Пропозиція співпраці новим гравцям
2	Поява нових технологій	Поява технологій, які дозволять оновити систему	Оновлення системи з використанням передових технологій
3	Зацікавленість зі сторони інших напрямків	Поява зацікавлених осіб в продуктах за межами сфери рітейла	Допрацювання продукту під потреби інших галузей

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – монополія/олігополія/ монополістична/чиста	Монополія	Повністю зайняти ринок до появи конкурентів
2. За рівнем конкурентної боротьби – локальний/національний/...	Світовий	Зайняти світовий ринок і стати лідером в даній сфері
3. За галузевою ознакою – міжгалузева/ внутрішньогалузева	Внутрішньогалузева	Пропонувати продукт найвищої якості, задовольняти побажання клієнта, оновлювати софт, з часом виходити на інші галузі
4. Конкуренція за видами товарів: – товарно–родова – товарно–видова – між бажаннями	товарно–видова	Пропонувати продукт найвищої якості, задовольняти побажання клієнта, оновлювати софт, з часом виходити
5. За характером конкурентних переваг – цінова / нецінова	Нецінова	Пропонувати продукт найвищої якості, задовольняти побажання клієнта, оновлювати софт, з часом виходити

Продовження таблиці 5.8

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
6. За інтенсивністю – марочна/не марочна	марочна	Пропонувати продукт найвищої якості, задовольняти побажання клієнта, оновлювати софт, з часом виходити

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки:	—	— є можливість наявності продукту — конкуренти відсутні	Не диктують.	Диктують. Є стандартні вимоги до продукту зв'язані з особливостями галузі	Замінники можуть з'явитись разом із конкурентами.

Виходячи з наведених вище даних визначимо перелік факторів конкурентоспроможності (Таблиця 5.10) та проведемо аналіз сильних та слабких сторін стартап-проекту (Таблиця 5.11).

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Першість на ринку	Відсутність конкурентів, повна монополія, можливість співпраці з усіма потенційними клієнтами
2	Високоякісний продукт	Єдине рішення, висока стабільність роботи
3	Широкі контакти в сфері рітейла	Хороші зв'язки з потенційними клієнтами

Складемо SWOT-аналіз (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (Таблиця 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін (Таблиця 5.11). На основі SWOT-аналізу розробимо альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок (Таблиця 5.13).

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін «SVA»

№ п/п	Фактор конкурентоспроможності	Бали 1–20	Рейтинг товарів–конкурентів у порівнянні з SVA						
			–3	–2	–1	0	+1	+2	+3
1	Першість на ринку	10	+						

Продовження таблиці 5.11

	Фактор конкурентоспроможності	Бали 1–20	Рейтинг товарів–конкурентів у порівнянні з SVA						
			–3	–2	–1	0	+1	+2	+3
2	Високоякісний продукт	10				+			
3	Широкі контакти в сфері ритейла	10				+			

Таблиця 5.12 – SWOT– аналіз стартап–проекту

Сильні сторони: Першість на ринку, Високоякісний продукт, Широкі контакти в сфері ритейла	Слабкі сторони:
Можливості: Розширення ритейл ринку, нові технології, зацікавленість інших галузей	Загрози: Поява конкурентів, Неможливість надавати якісний продукт

Таблиця 5.13 – Альтернативи ринкового впровадження стартап–проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Об'єднання з ведучими гравцями ринку ритейла	90%	6 місяців

5.4 Розроблення ринкової стратегії проекту

Розглянемо основні цільові групи потенційних споживачів (Таблиця 5.14). Виходячи з їх аналізу цільовими групами буде обрано наступні групи: FMCG,

Дрогері, АЗС, ДІУ. Сформуємо базову стратегію розвитку (Таблиця 5.15) та виберемо стратегії конкурентної поведінки (Таблиця 5.16).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	FMCG	100%	високий	відсутня	дуже легко
2	Дрогері	100%	високий	відсутня	дуже легко
3	Електроніка	15% – 30%	низький	відсутня	важко
4	АЗС	75%	середній	відсутня	легко
5	Одяг	0% – 15%	низький	відсутня	дуже важко
6	ДІУ	50% – 100%	середній	відсутня	легко

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Об'єднання з ведучими гравцями ринку рітейла	Монополізація	монополія	Стратегія спеціалізації

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Так	Шукати нових	Конкуренти відсутні	Стратегія зайняття конкурентної ніші

Розробимо стратегію позиціонування (Таблиця 5.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати проект.

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап–проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Якісний продукт, який виконує заявлені функції	Задоволення потреб клієнтів	Першість на ринку, високоякісний продукт, зв'язки	Лідерство, якість, точність

5.5 Розроблення маркетингової програми стартап–проекту

Сформуємо маркетингову концепцію товару, який отримає споживач. Для цього підсумуємо результати попереднього аналізу конкурентоспроможності товару (Таблиця 5.18) та розробимо трирівневу маркетингову модель товару (Таблиця 5.19). Товар буде захищений від копіювання за рахунок неможливості отримання доступу до вихідного коду.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Підтримання встановленого асортименту магазину в якісній і кількісній складових.	Можливість контролювати повноту і точність асортименту магазину, що в свою чергу приводить до підвищення прибутку. Зменшення витрат на найм персоналу, мінімізація людського фактору.	Конкуренти відсутні

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Потреба Підтримання встановленого асортименту магазину в якісній і кількісній складових. Надає можливість контролювати повноту і точність асортименту магазину, що в свою чергу приводить до підвищення прибутку. Зменшення витрат на найм персоналу, мінімізація людського фактору.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. SaaS	—	—
	Якість: тестування показало якість ефективної підтримки асортименту на рівні 97%		
	Пакування —		
	Марка: SVA		
III. Товар із підкріпленням	—		
	—		

Визначимо цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар. Проаналізуємо ціни на товари-аналоги або товари субституту, а також рівень доходів цільової групи споживачів (Таблиця 5.20). Також визначимо оптимальну систему збуту (Таблиця 5.21).

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари– замінники	Рівень цін на товари– аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	–	–	300 млн – 55 млрд в рік	5000 \$ впровадження + 1000 \$ щомісячна підтримка Верхня: 250 000 \$ впровадження + 10000\$ щомісячна підтримка

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	SaaS	Впровадження; надання підтримки;	Глибокий	Встановлений контакт з клієнтом.

Розробимо концепцію маркетингових комунікацій, що спираються на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (Таблиця 5.22)

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/ п	Специфік а поведінки цільових клієнтів	Канали комунікацій, якими користуютьс я цільові клієнти	Ключові позиції, обрані для позиціонуванн я	Завдання рекламного повідомленн я	Концепція рекламного звернення
1	Бажання отримати максимум за мінімальн у ціну	Прямий контакт	Підтримка асортименту магазину	Запевнити клієнта, що продукт принесе йому value	Замінімо мерчандайзері в в рітейлі

Висновки за розділом 5

У даному розділі було проведено глибоке дослідження ринку і потенційних можливостей та загроз для розвитку продукту.

Згідно з наведеними таблицями, можемо зробити висновки, що у продукту буде попит, так як він має унікальну цінність. Продукт буде задавати ринок, тому треба буде бути готовими до появи сильних противників та активізації уже існуючих конкурентів для ефективної боротьби необхідно оперативно захопити всю цільову групу клієнтів.

У продукту є перспективи впровадження з огляду на потенційні групи клієнтів. Активної конкуренції поки що немає, проте уже є компанії, які активно займаються розробкою подібного програмного забезпечення. А також існують великі гравці в сфері софту для рітейлу, які можуть захотіти перешкодити виходу

продукту на ринок. Якщо не встигнути охопити весь ринок до виходу на нього конкурентів, доведеться зайняти стратегію кооперації з рітейл бізнесом.

Подальша імплементація продукту є доцільною.

ВИСНОВКИ

В даній магістерській дисертації було реалізовано систему класифікації товарів для інтелектуальної системи мерчандайзингу, яка дозволяє стабільно класифікувати товари. Розробка такої системи є актуальною та перспективною задачею на сьогоднішній день, так як ефективних і доступних систем мерчандайзингу на ринку поки що ще немає.

Для реалізації поставленої задачі були виконані наступні кроки:

1. досліджено та проаналізовано існуючі підходи до класифікації товарів;
2. досліджено основні структурні елементи системи класифікації товарів, а саме: структури згорткових нейронних мереж, сучасні функції активації, техніку Dropout, методи оптимізації функції ціни, види функцій ціни;
3. запропоновано та аргументовано використання унікальної структури згорткової нейронної мережі;
4. запропоновано та аргументовано використання бінарної кросс-ентропії як цінової функції, так як така функція ідеально підходить для задачі бінарної класифікації;
5. запропоновано алгоритм навчання класифікатора і класифікації товарів;
6. розроблено програмну реалізацію запропонованого алгоритму класифікації товарів;
7. розроблено інтерфейс, який дає змогу користувачу працювати з інтелектуальною системою мерчандайзингу в вигляді персонального кабінету користувача з розширеним функціоналом;
8. аргументовано використання мови програмування Python і СУБД MySQL;

9. оцінено складність алгоритму розробленої системи класифікації товарів і проведено порівняння якості роботи запропонованої мережі з загальновідомими;
10. приведено опис роботи інтелектуальної системи мерчандайзингу в реальних умовах;
11. представлено рекомендації щодо напрямків подальших вдосконалень.

Було розглянуто сфери застосування систем мерчандайзингу та аргументовано актуальність використання системи. Проаналізовано існуючі методи, алгоритми та підходи до автоматизованого мерчандайзингу.

Досліджено структуру системи класифікації товарів. Проведено аналіз структурних елементів системи, а саме:

- згорткових шарів в CNN;
- об'єднувальних шарів в CNN;
- повнозв'язних шарів в CNN;
- поняття тензора;
- функції активації ReLU;
- функції активації сигмоїд;
- техніки Dropout;
- цінової функції бінарна кросс-ентропія;
- методу оптимізації цінової функції;
- навчальної, тестової і валідаційної вибірки.

Описано алгоритми, що використовуються для реалізації системи класифікації товарів.

Запропоновано алгоритм класифікації товарів, структуру розробленої нейронної мережі. Це дозволяє ефективно проводити бінарну класифікацію, зберігати ваги для кожного товару окремо, точно визначати клас товару в одній товарній групі. Запропонована структура значно зменшує необхідні ресурси для навчання і дозволяє ефективно працювати з великими об'ємами даних.

Аргументовано використання запропонованого методу оптимізації, який дозволяє прискорити процес навчання мережі.

Також було обґрунтовано вибір мови програмування для реалізації системи класифікації товарів. Досліджено та проаналізовано запропонований алгоритм роботи системи класифікації товарів. Описано та проілюстровано роботу запропонованої системи.

Практичним результатом магістерської дисертації є інтелектуальна система мерчендайзингу у вигляді програмного продукту, що дозволяє проводити аналіз полиць в реальному часі, з точністю 96% визначати наявність та відповідність планограмі представленого товару та дозволяє накопичувати історію наповненості полиць для подальшого статистичного аналізу, а також реалізація системи класифікації товарів як окремого модулю інтелектуальної системи мерчендайзингу, яка дозволяє класифікувати товари з точністю 95-97% і є невідомою частиною загальної системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Эстерлинг С. Мерчандайзинг в розничной торговле / Эстерлинг С, Флоттман Э., Джернинган М., Марал С. – СПб. : Питер, 2004. – 304 с.
2. Червак Р. Мерчандайзинг: искусство тореадора. / Червак Р., Бондарчук В., Синило Л. - К. : Изд-во Алексея Капусты, 2003. – 306 с.
3. Сысоева С.В. Мерчандайзинг. Курс управления ассортиментом в рознице / Снегирева В.В., Бузукова Е. А. – СПб. : Питер, 2008. – 226 с.
4. Bohyung Han. Density-Based Multifeature Background Subtraction With Support Vector Machine / Bohyung Han and Larry S. Davis // IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 34, No. 5. – 2012. – pp. 33 – 38.
5. Zhu Xiaojin. Supervised learning literature survey / Zhu Xiaojin. – Department of Computer Science and Engg, University of Wiskonsin-Maddison, 2005. – 1530 p.
6. Jipsa Kurian. A survey on image classification methods / Jipsa Kurian and V. Karunakaran // Karunya University India, IJARECE, Vol. 1, Issue 4. – 2012. – pp. 69 – 72.
7. Jianxin Wu. Efficient Hik SVM Learning For Image Classification / Jianxin W // IEEE Transactions On Image Processing, Vol. 21, No. 10. – 2012. – pp. 45 – 57.
8. Vijay Kumar. Importance of Statistical Measures in Digital Image Processing / Vijay Kumar, Priyanka Gupta // International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Vol. 2, Issue 8. – 2012. – pp. 71 – 77.
9. S. Arunadevi, A Survey On Image Classification Algorithm Based On Per-Pixel / S. Arunadevi, dr. S. Daniel Madan Raja // International Journal Of Engineering Research And General Science, Vol 2, Issue 6. – 2014. – pp. 12 – 25.

10. Taherkhani. Recognizing sorting algorithms with the C4.5 decision tree classifier in Program Comprehension (ICPC) / A. Taherkhani // 2010 IEEE 18th International Conference – 2010. – pp. 72-75.
11. Y. Xu. MDCT based 3-D texture classification of emphysema and early smoking related lung pathologies / Y. Xu, M. Sonka, G. McLennan, J. Guo, and E. A. Hoffman / IEEE Trans. Med. Imag. Vol. 25, No. 4. – 2006. – pp. 464–475.
12. L. Sorensen. Quantitative analysis of pulmonary emphysema using local binary patterns / L. Sorensen, S. B. Shaker, and M. de Bruijne // IEEE Trans. Med. Imag. Vol. 29, No. 2. – 2010. – pp. 559–569.
13. Jian Xin Wu. Efficient Hik SVM Learning For Image Classification / Jian Xin Wu // IEEE Transactions On Image Processing, Vol. 21, No. 10. – 2012. – pp. 9 – 12.
14. Lizhen Lu. A Decision-Tree Classifier for Extracting Transparent Plastic-Mulched Land cover from Landsat-5 TM Images / Lizhen Lu, Liping Di, Senior and Yanmei Ye // IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing, Vol. 7, No. 11. – 2014. – pp. 3 – 7.
15. LeCun Y. Gradient-based learning applied to document recognition / LeCun Y., Bottou L., Bengio Y., and Haffner P. // Proceedings of the IEEE. – 1998. – pp. 2278–2324.
16. Ian Goodfellow. Deep Learning / Ian Goodfellow, Yoshua Bengio and Aaron Courville. – Boston: The MIT Press, 2016. – 800 p.
17. LeCun Y. Deep learning / LeCun Y., Bengio Y. and Hinton G. // Nature, 521(7553). – 2015. – pp. 436 – 444.
18. Yu D. Mixed pooling for convolutional neural networks / Yu D., Wang H., Chen P. and Wei, Z. // Proceedings of the 9th International Conference on Rough Sets and Knowledge Technology. – 2014. – pp. 364–375.
19. Krizhevsky A. ImageNet classification with deep convolutional neural networks / Krizhevsky A., Sutskever I. and Hinton, G. E. // Advances in neural information processing systems, 25. – 2012. – pp. 1097–1105.

20. Szegedy C. Going deeper with convolutions / Szegedy C. and Liu W. // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – pp. 1–9.
21. Zeiler M. D. Visualizing and understanding convolutional networks / Zeiler M. D. and Fergus R. // Proceedings of the European Conference on Computer Vision. – 2014. – pp. 818–833.
22. Deng L. Deep learning: Methods and applications / Deng L. and Yu D. // Foundations and Trends in Signal Processing, 7(3–4) – 2014. – pp. 197–387.
23. Hinton G. E. Reducing the dimensionality of data with neural networks / Hinton G. E. and Salakhutdinov R. R. // Science, 313(5786). – 2006. – pp. 504–507.
24. Duchi J. Adaptive subgradient methods for online learning and stochastic optimization / Duchi J., Hazan E. and Singer Y. // Journal of Machine Learning Research, 12. – 2011. – pp. 2121–2159.
25. RMSProp. [Электронный ресурс] / Tieleman T. and Hinton G // COURSERA: Neural Networks for Machine Learning. – 2012. – Режим доступа: <https://www.coursera.org/learn/deep-neural-network/lecture/BhJlm/rmsprop>.
26. Adam: A Method for Stochastic Optimization. [Электронный ресурс] / D.P Kingma and L.J. Ba // ICLR. – 2015. – Режим доступа: <https://arxiv.org/pdf/1412.6980.pdf>.
27. He K. Deep Residual Learning for Image Recognition / He K., Zhang X., Ren S., Sun J. // Proceedings of the CVPR 2016. – 2016. – pp. 770–778.
28. Kawano Y. Food Image Recognition with Deep Convolutional Features / Kawano Y., Yanai K. // Proceedings of the UbiComp. – 2014. – pp. 589–593.
29. Yanai K. Food Image Recognition using Deep Convolutional Network with Pre-Training and Fine-Tuning / Yanai K. Kawano Y. // Proceedings of the ICMEW. – 2015. – pp. 1–6.
30. Singla A. Food/Non-Food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model / Singla A. Yuan L. Ebrahimi T. // Proceedings of the MADiMa'16. – 2016. – pp. 3–11.

- 31.Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.

ДОДАТОК А ІЛЮСТРАТИВНИЙ МАТЕРІАЛ ДОПОВІДІ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу

Магістерська дисертація

на тему: “Інтелектуальна система мерчандайзингу”

Виконали:
Студенти групи КА-62м
Литвинюк Антон Андрійович
Левчук Святослав Богданович

Науковий керівник:
к.т.н., доцент
Дідковська М.В.

Київ - 2018

Актуальність роботи

2

Динамика розничного товарообороту підприємств України, млрд грн



Источник: Государственная служба статистики Украины

Актуальність роботи

3

ІТ системи для ритейла:

- лояльність;
- ціноутворення;
- прогнозування продаж;
- автозамовлення.

Ключові задачі для розвитку в ритейлі:

- забезпечення доступності товару на полиці;
- розширення аудиторії покупців;
- зменшення списання товарів та товарних залишків;
- встановлення оптимальних цін.

Актуальність роботи

4

Статистики від ECR:

- в середньому всього 20% позицій в статусі OOS поповнюється протягом 8 годин;
- 8,3%, середній показник OOS по індустрії в світі, не змінюється останні 8 років;
- підвищення OSA на 3% принесе ~ 1% зростання обсягів поставок для постачальника;
- підвищення OSA на 2% принесе ~ 1% зростання обсягів продажів для ритейлера;
- у 47% випадків при відсутності потрібного йому товару на полиці покупець не зробить покупку в цьому магазині;
- у 46% випадків при відсутності на полиці магазину потрібного йому товару покупець не зробить покупку товару даного виробника

Існуючі рішення

5

**ShelfMatch****Trax**
image recognition**MOBI.COM****SMART
SHELF
SOLUTIONS****Intelligence**
Retail

Постановка задачі

6

- Дослідити та провести аналіз існуючих систем мерчандайзингу;
- Дослідити та провести аналіз існуючих систем класифікації товарів ;
- Дослідити та провести аналіз існуючих систем сегментації та маппінгу;
- Розробити систему сегментації та маппінгу асортименту відносно планограми;
- Розробити систему класифікації товарів на полицях;
- Розробити систему мерчандайзингу.

Мета, об'єкт, предмет дослідження

7

Мета роботи:

Розробка та дослідження інтелектуальної системи мерчендайзингу.

Об'єкт дослідження:

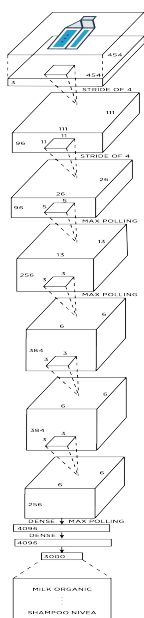
Методи мерчендайзингу товарів в торгових точках.

Предмет дослідження:

Методи сегментації та класифікації товарів на полицях в торгових точках.

Глибока згорткова мережа на базі AlexNet

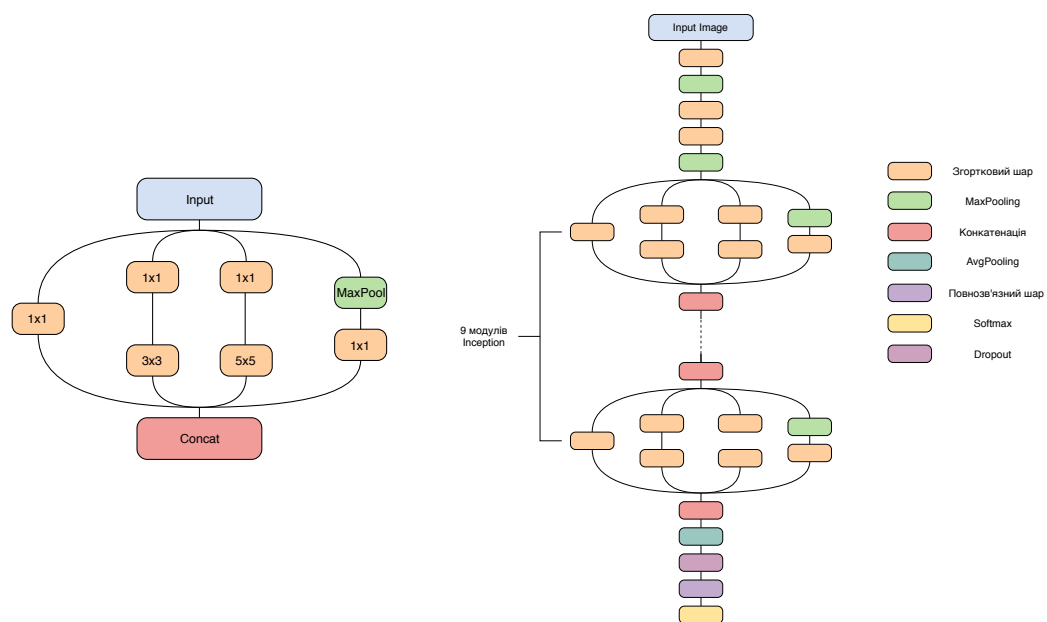
8



Модель	Тренувальний сет		Валідаційний сет		Тестовий сет	
	Loss	Acc.	Loss	Acc.	Loss	Acc.
AlexNet SGD	0.41	89.78	0.55	81.99	0.44	84.77
AlexNet NAG	0.33	89.90	0.57	82.04	0.43	84.12
AlexNet AdaGrad	0.52	89.37	0.61	80.20	0.46	84.89
GoogLeNet SGD	0.41	90.70	0.80	80.64	0.59	86.33
GoogLeNet NAG	0.36	90.73	0.79	80.66	0.59	86.12
GoogLeNet AdaGrad	0.48	87.50	0.75	81.24	0.47	86.64
ResNet SGD	0.64	81.86	0.34	85.42	0.28	87.77
ResNet NAG	0.43	83.88	0.27	85.08	0.26	87.99
ResNet AdaGrad	0.51	83.21	0.33	83.78	0.32	86.51
Proposed SGD	0.45	88.59	0.45	80.71	0.26	86.69
Proposed NAG	0.44	88.53	0.43	81.12	0.26	86.58
Proposed AdaGrad	0.44	89.76	0.45	80.65	0.25	87.48

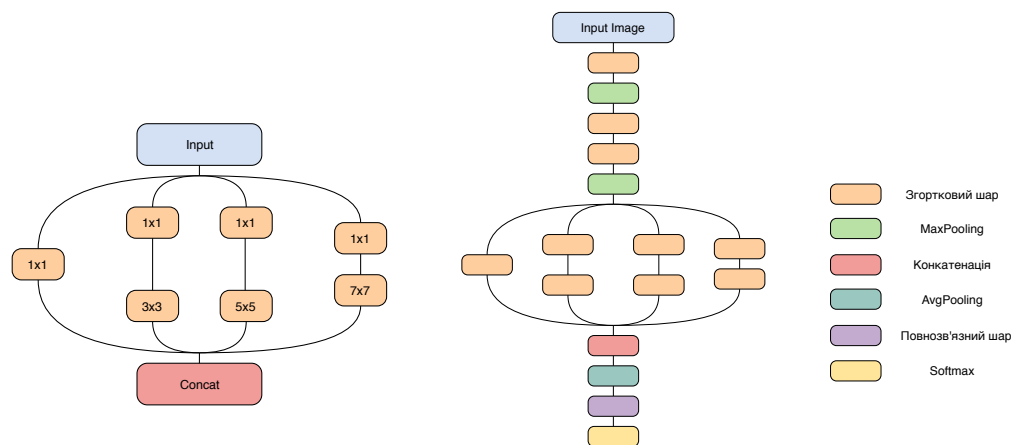
GoogLeNet Inception

9



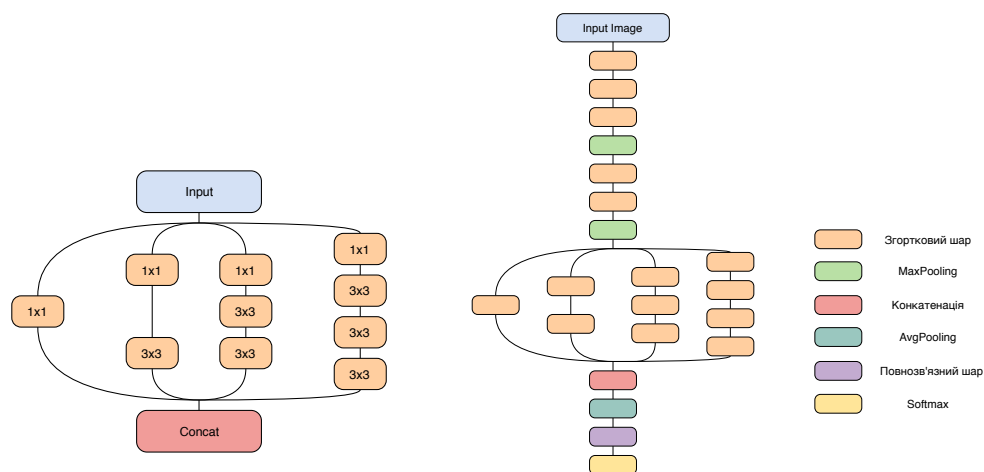
Запропонована мережа на базі GoogLeNet Inception

10



Запропонована мережа на базі GoogLeNet Inception

11



Глибока згорткова мережа на базі GoogLeNet Inception

12

Модель	Точність	Швидкість (зображень/мс)	Кількість параметрів (млн)
AlexNet	86.78	13.8	58.48
GoogLeNet	91.94	40.36	10.45
Proposed ver.1	92.07	18.21	0.87
Proposed ver.2	92.89	37.48	0.88

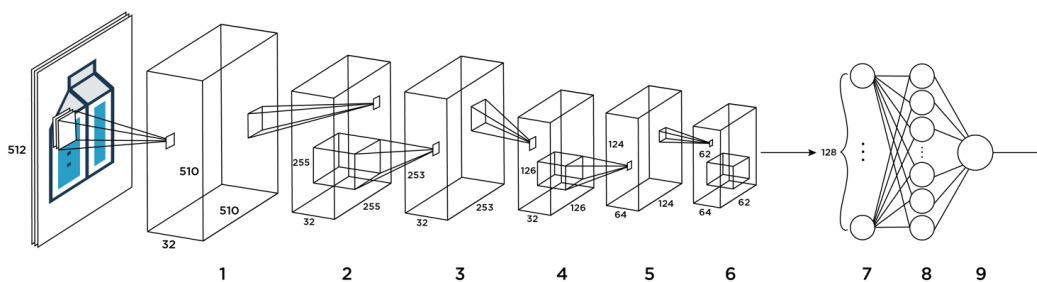
Запропонований класифікатор

13

Model	Avg. Prob.	Pos.	Neg.	Avg. Acc.
32c+p – 32c+p – 128fc – 1fc	66.9	88	12	88%
32c+p – 32c+p – 64c+p – 128fc+dropout – 1fc	75.1	96	4	96%
32c+p – 32c+p – 32c+p – 64c+p – 128fc – 1fc	69.4	86	14	86%
32c+p – 32c+p – 32c+p – 64c+p – 64c+p – 128fc – 1fc	68.2	89	11	89%
32c+p – 32c+p – 32c+p – 64c+p – 128fc+dropout – 1fc	73.89	92	8	92%
32c+p – 32c+p – 32c+p – 64c+p – 64c+p – 128fc+dropout – 1fc	75.47	93	7	93%

Запропонований класифікатор

14



Датасети

15

Клас "object":

- 1000 зображень, на яких зображено товар, який класифікується.

Клас "not object":

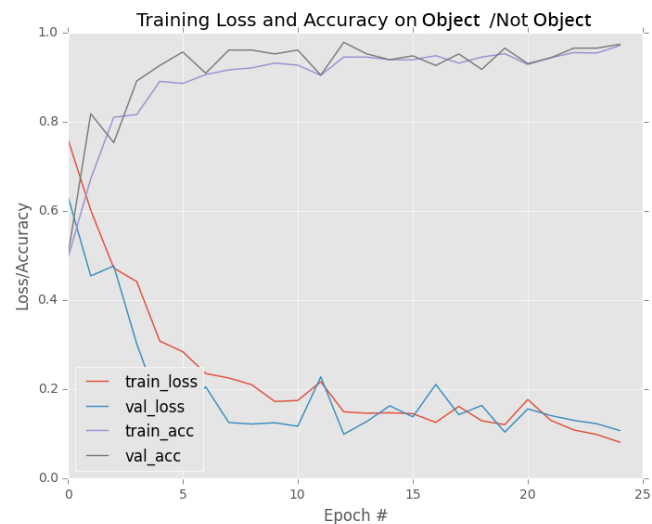
- 500 зображень з найпоширенішими товарами в ритейлі;
- 500 зображень з іншими товарами з товарної групи об'єкта.

Розбиття вибірки на навчальну/валідаційну/тестову:

- 70% навчальна;
- 15% валідаційна;
- 15% тестова.

Аналіз якості навчання

16



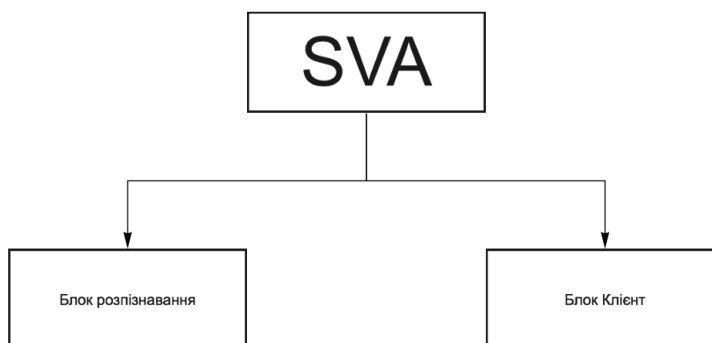
Аналіз якості класифікації

17

Group	Avg. Prob., %	Positive Count	Negative Count	Avg. Acc., %
Молоко	69.8	5	0	100
Йогурти	75.4	5	0	100
Пиво	73.2	5	0	100
Сидр	73.1	5	0	100
Кава	82.7	5	0	100
Чай	63.4	3	2	60
Крупи	65.9	5	0	100
Соки	88.3	5	0	100
Цукор	71.2	5	0	100
Сіль	72.3	5	0	100
Мука	69.9	4	1	80
Вода сол.	83.4	5	0	100
Горілка	84.6	5	0	100
Консерви	81.2	5	0	100
Шоколад	68.7	5	0	100
Конд. Фас.	60.3	4	1	80
Сухі снід.	91.4	5	0	100
Макарони	73.6	5	0	100
Вода мін.	88.12	5	0	100
Цукерки	65.3	5	0	100
Σ	75.1	96	4	96

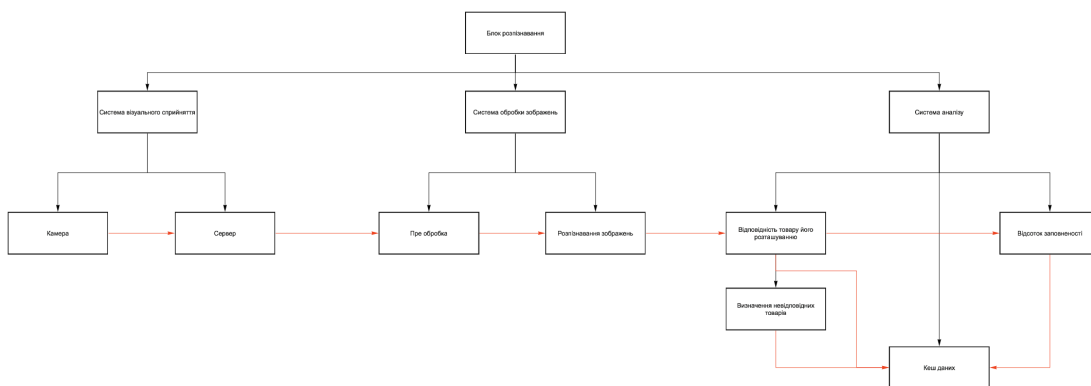
Запропонована інтелектуальна система мерчандайзингу

18



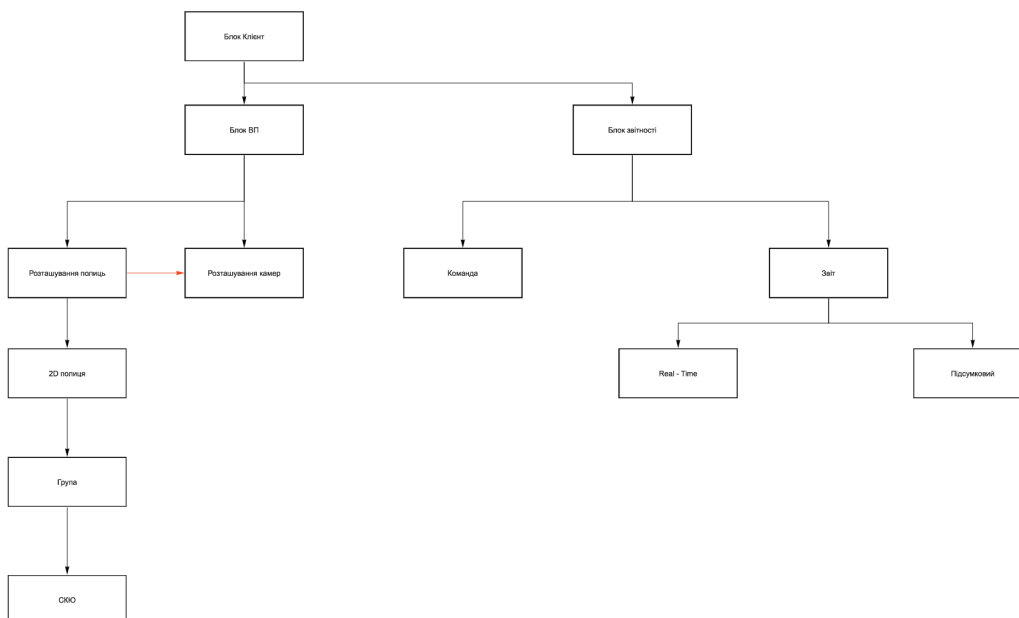
Запропонована інтелектуальна система мерчандайзингу

19



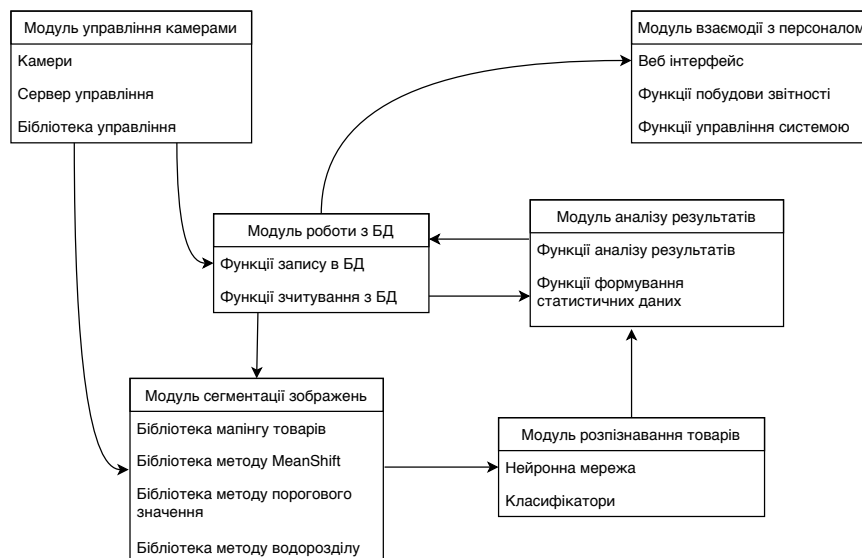
Запропонована інтелектуальна система мерчандайзингу

20



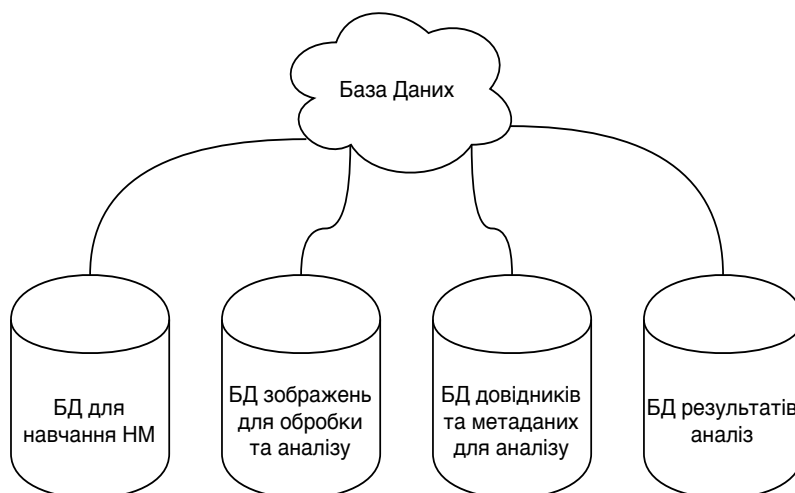
Архітектура системи

21



Архітектура БД

22



Аналіз результатів розробки

23

- досліджено існуючі системи мерчандайзингу;
- проаналізовано методи сегментації зображень;
- запропоновано та аргументовано використання комбінованого методу сегментації зображення, в умовах практичної задачі;
- описано алгоритм роботи запропонованого методу з використанням динамічного вікна, в умовах практичної задачі;
- проаналізовано підходи до класифікації товарів;
- запропоновано та аргументовано використання унікальної структури згорткової нейронної мережі;
- запропоновано алгоритм навчання класифікатора і класифікації товарів;
- розроблено програмну реалізацію інтелектуальної системи мерчандайзингу;
- оцінено точність роботи розробленої системи сегментації та маппінгу асортименту, яка складає 96 – 98%;
- оцінено точність роботи розробленої системи класифікації асортименту, яка складає 95 – 97%;
- оцінено якість роботи розробленої інтелектуальної системи мерчандайзингу, яка складає 95 – 97%

Практичні результати роботи системи

24


[MAIN](#)
[REPORTS](#)
[LOG](#)
[ABOUT](#)
[LOG IN](#)
[All](#)
[Region](#)
[Store](#)
 Search

1. Kyiv
2. Lviv
3. Dnipro
4. Kharkiv
5. Odessa

Наукова новизна

25

Наукова новизна полягає в:

- Запропонованому підході до автоматизації мерчандайзингу;
- Запропонованому комбінованому методі сегментації та мапінгу, що базується на планограмі ТТ та дозволяє виділити товари на полиці для їх аналізу;
- Запропонованому підході до класифікації товарів, що базується на спеціально розробленій глибокій згортковій нейронній мережі для визначення відповідності розташованого товару на полиці до планограми.

Практична цінність

26

Реалізовано інтелектуальну систему мерчандайзингу для ритейла у вигляді програмного продукту, яка дозволяє:

- Аналізувати полиці в ТТ в реальному часі з точністю 96%;
- Накопичувати історію наповненості полиць для подальшого статистичного аналізу;
- Підтримувати якість викладки товарів на полиці;
- Забезпечувати наявність товарів на полиці;
- Проводити моніторинг якості роботи персоналу торгового залу;
- Автоматизувати процес мерчандайзингу.

Шляхи подальшого розвитку

27

- Додавання нового функціоналу для збору статистичних даних;
- Додавання нового функціоналу для більш глибокого аналізу полиці;
- Розробка функціоналу для аналізу поведінки покупців;
- Покращення інтерфейсу користувача;
- Імплементация новітніх підходів до класифікації, по мірі їх появи на ринку;
- Кооперації з іншими IT рішеннями для ритейлу;
- Залучення нових клієнтів в сфері ритейл бізнесу.

Дякуємо за увагу!

ДОДАТОК Б ЛІСТИНГ

Б1. Програмний код класу Conv

```
class Conv(_Conv):

    def __init__(self, filters,
                  kernel_size,
                  strides=(1, 1),
                  padding='valid',
                  data_format=None,
                  dilation_rate=(1, 1),
                  activation=None,
                  use_bias=True,
                  kernel_initializer='glorot_uniform',
                  bias_initializer='zeros',
                  kernel_regularizer=None,
                  bias_regularizer=None,
                  activity_regularizer=None,
                  kernel_constraint=None,
                  bias_constraint=None,
                  **kwargs):
        super(Conv2D, self).__init__(
            rank=2,
            filters=filters,
            kernel_size=kernel_size,
            strides=strides,
            padding=padding,
            data_format=data_format,
            dilation_rate=dilation_rate,
            activation=activation,
            use_bias=use_bias,
            kernel_initializer=kernel_initializer,
            bias_initializer=bias_initializer,
            kernel_regularizer=kernel_regularizer,
            bias_regularizer=bias_regularizer,
            activity_regularizer=activity_regularizer,
            kernel_constraint=kernel_constraint,
```

```

        bias_constraint=bias_constraint,
        **kwargs)
self.input_spec = InputSpec(ndim=4)

def get_config(self):
    config = super(Conv2D, self).get_config()
    config.pop('rank')
    return config

```

Б2. Программный код класу Pool

```

class Pool(_Pooling2D):

def __init__(self, pool_size=(2, 2), strides=None, padding='valid',
             data_format=None, **kwargs):
    super(MaxPooling2D, self).__init__(pool_size, strides, padding,
                                       data_format, **kwargs)

def _pooling_function(self, inputs, pool_size, strides,
                     padding, data_format):
    output = K.pool2d(inputs, pool_size, strides,
                     padding, data_format,
                     pool_mode='max')
    return output

```

Б3. Программный код класу Optim

```

class Optim(Optimizer):

def __init__(self, lr=0.001, beta_1=0.9, beta_2=0.999,
             epsilon=None, decay=0., amsgrad=False, **kwargs):
    super(Adam, self).__init__(**kwargs)
    with K.name_scope(self.__class__.__name__):
        self.iterations = K.variable(0, dtype='int64', name='iterations')

```



```

self.lr = K.variable(lr, name='lr')
self.beta_1 = K.variable(beta_1, name='beta_1')
self.beta_2 = K.variable(beta_2, name='beta_2')
self.decay = K.variable(decay, name='decay')
if epsilon is None:
    epsilon = K.epsilon()
self.epsilon = epsilon
self.initial_decay = decay
self.amsgrad = amsgrad

@interfaces.legacy_get_updates_support
def get_updates(self, loss, params):
    grads = self.get_gradients(loss, params)
    self.updates = [K.update_add(self.iterations, 1)]

    lr = self.lr
    if self.initial_decay > 0:
        lr *= (1. / (1. + self.decay * K.cast(self.iterations,
                                              K.dtype(self.decay))))

    t = K.cast(self.iterations, K.floatx()) + 1
    lr_t = lr * (K.sqrt(1. - K.pow(self.beta_2, t)) /
                (1. - K.pow(self.beta_1, t)))

    ms = [K.zeros(K.int_shape(p), dtype=K.dtype(p)) for p in params]
    vs = [K.zeros(K.int_shape(p), dtype=K.dtype(p)) for p in params]
    if self.amsgrad:
        vhots = [K.zeros(K.int_shape(p), dtype=K.dtype(p)) for p in params]
    else:
        vhots = [K.zeros(1) for _ in params]
    self.weights = [self.iterations] + ms + vs + vhots

    for p, g, m, v, vhat in zip(params, grads, ms, vs, vhots):
        m_t = (self.beta_1 * m) + (1. - self.beta_1) * g
        v_t = (self.beta_2 * v) + (1. - self.beta_2) * K.square(g)
        if self.amsgrad:
            vhat_t = K.maximum(vhat, v_t)
            p_t = p - lr_t * m_t / (K.sqrt(vhat_t) + self.epsilon)
            self.updates.append(K.update(vhat, vhat_t))
        else:
            p_t = p - lr_t * m_t / (K.sqrt(v_t) + self.epsilon)

```

```

self.updates.append(K.update(m, m_t))
self.updates.append(K.update(v, v_t))
new_p = p_t

# Apply constraints.
if getattr(p, 'constraint', None) is not None:
    new_p = p.constraint(new_p)

self.updates.append(K.update(p, new_p))
return self.updates

def get_config(self):
    config = {'lr': float(K.get_value(self.lr)),
              'beta_1': float(K.get_value(self.beta_1)),
              'beta_2': float(K.get_value(self.beta_2)),
              'decay': float(K.get_value(self.decay)),
              'epsilon': self.epsilon,
              'amsgrad': self.amsgrad}
    base_config = super(Adam, self).get_config()
    return dict(list(base_config.items()) + list(config.items()))

```

Б4. Программный код Train

```

train_dir = 'train'

val_dir = 'val'

test_dir = 'test'

img_width, img_height = 150, 150

input_shape = (img_width, img_height, 3)

epochs = 45

batch_size = 16

nb_train = 700

```

```

nb_val = 150

nb_test = 150

if K.image_data_format() == 'channels_first':
    input_shape = (3, img_width, img_height)
else:
    input_shape = (img_width, img_height, 3)

model = Sequential()

#Recognition layers
model.add(Conv2D(32, (3, 3), input_shape=input_shape))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

#Classification layers
model.add(Flatten())
model.add(Dense(128))
model.add(Activation('relu'))
model.add(Dropout(0.5))
model.add(Dense(1))
model.add(Activation('sigmoid'))

plot_model(model, to_file='model_plot.png', show_shapes=True, show_layer_names=True)

# Compile network
model.compile(loss='binary_crossentropy', #Classification algorithm
              optimizer='adam',          #Optimization algorithm
              metrics=['accuracy'])

# Suitable image generation
train_datagen = ImageDataGenerator(
    rescale=1. / 255,

```

```

shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1. / 255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

val_generator = train_datagen.flow_from_directory(
    val_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='binary')

history= model.fit_generator(
    train_generator,
    steps_per_epoch=nb_train // batch_size,
    epochs=epochs,
    validation_data=val_generator,
    validation_steps=nb_val // batch_size)

plot_model(history)

scores = model.evaluate_generator(test_generator, nb_test // batch_size)

print("Accuracy on test data: %.2f%%" % (scores[1]*100))

model.save('classy_model_results.h5')
model.save_weights('classy_weights_results.h5')

```